

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN
PELAPORAN SARANA DAN PRASARANA STUDI PADA
FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Danniar Reza Firdausy
NIM: 145150401111063



PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
TAHUN
2018

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PELAPORAN SARANA DAN PRASARANA STUDI PADA FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA

SKRIPSI

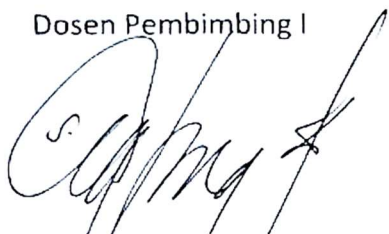
Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Danniar Reza Firdausy
NIM: 145150401111063

Skripsi ini telah diuji dan dinyatakan lulus pada
2 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Satrio Agung Wicaksono, S.Kom, M.Kom
NIP: 19860521 201212 1 001

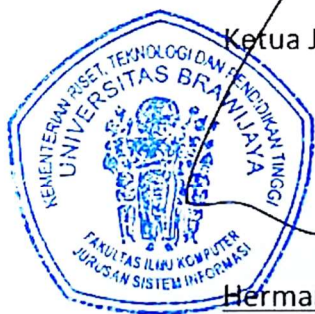
Dosen Pembimbing II



Fajar Pradana, S.ST, M.Eng
NIP: 19871121 201504 1 004

Mengetahui

Ketua Jurusan **Sistem Informasi**



Herman Tolle, Dr. Eng., S.T, M.T
NIP: 19740823 200012 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 2 Juli 2018



Danniar Reza Firdausy

NIM: 145150401111063

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana Studi pada Fakultas Ilmu Komputer Universitas Brawijaya”

Penelitian ini tidak dapat penulis selesaikan tanpa didukung oleh ilmu yang didapatkan di kelas perkuliahan dan di saat yang sama juga mengajarkan banyak ilmu lain yang belum penulis dapatkan di kelas perkuliahan. Selain itu, penelitian ini juga memberikan penulis pengalaman untuk dapat berkomunikasi dengan banyak orang, menggali permasalahan yang ada, menyamakan persepsi yang penulis miliki dengan civitas akademik Fakultas Ilmu Komputer dan pemangku kepentingan lain serta menyamakan persepsi dengan kedua dosen pembimbing. Proses penelitian yang dilakukan juga mengajarkan penulis untuk menjadi pribadi yang sabar dan senantiasa berbagi hal-hal bermanfaat yang berkaitan dengan kebutuhan pembelajaran. Penelitian ini tidak dapat diselesaikan tanpa dukungan dan bantuan dari orang-orang terdekat, sehingga saya ingin menyampaikan rasa terima kasih kepada:

1. Bapak Satrio Agung Wicaksono , S.Kom, M.Kom, selaku pembimbing satu yang telah memberikan kesempatan kepada penulis untuk menjadi mahasiswa bimbingan skripsinya dan melakukan penelitian dengan studi kasus yang telah beliau berikan, serta menjadi pembimbing yang sabar, solutif dan senantiasa memberikan contoh sebagai pribadi yang professional dalam membimbing penyelesaian skripsi penulis.
2. Bapak Fajar Pradana, S.ST, M.Eng, selaku pembimbing dua yang telah memberikan kesempatan dan waktunya untuk membimbing penulis dalam menyelesaikan penelitian yang dilakukan, serta senantiasa dengan sabar dan teliti dalam memeriksa pengerjaan dokumen skripsi sampai terselesaikannya skripsi ini.
3. Bapak Aryo Pinandito, S.T, M.MT, selaku pembimbing akademik yang telah meluangkan waktunya untuk memberikan bimbingan perihal akademik dari penulis selama berjalannya perkuliahan, sehingga penulis mampu menjalani perkuliahan sesuai dengan prosedur yang telah ditetapkan oleh akademik FILKOM.
4. Bapak Ferix Panji Andrianto, S.ST, selaku pegawai Perlengkapan FILKOM yang telah bersedia untuk menjadi narasumber, menguji sistem informasi yang telah dikembangkan serta memberikan masukan-masukannya terhadap pengembangan dari sistem informasi ini.
5. Ibu saya, Wahyuni Sulistyaningsih yang selalu dengan sabar memberikan dukungan berupa doa, moril dan materiil agar penulis senantiasa termotivasi untuk menyelesaikan penelitian yang sedang dilakukan. Ayah

saya, Fuad Achmadi yang senantiasa memberikan dukungan moril, materiil, doa, nasihat dan motivasi sehingga penulis mampu menyelesaikan penelitian ini.

6. Rosalia Indah, selaku sahabat penulis yang telah dengan setia, sabar, penuh perhatian mendampingi penulis serta saling menyemangati selama berjalannya perkuliahan, kepengurusan lembaga mahasiswa, kepengurusan asisten praktikum basis data dan pengerjaan penelitian skripsi. Edwin Nurwansyah yang telah senantiasa menjadi teman diskusi dalam pengerjaan skripsi dan dalam mengembangkan aplikasi *Android*. Firnanda Iftitah Dewi Antono dan Nanda Adhi Winata yang juga telah menjadi teman untuk diskusi dan berbagi ilmu dalam pengerjaan skripsi.
7. Teman-teman Lembaga Badan Perwakilan Mahasiswa Sistem Informasi dan Eksekutif Mahasiswa Sistem Informasi yang telah menjadi sarana bagi penulis untuk berkembang menjadi pribadi yang solutif, peka terhadap permasalahan di sekitar dan aktif berorganisasi di kalangan masyarakat.
8. Teman-teman Asisten Praktikum Basis Data periode 2016-2017 dan periode 2017-2018 yang telah menjadi sarana bagi penulis untuk berkembang menjadi pribadi yang gemar untuk berdiskusi, saling berbagi ilmu yang dimiliki serta gemar untuk bekerja dan belajar bersama.
9. Teman-teman mahasiswa Jurusan Sistem Informasi dan jurusan lainnya dari Fakultas Ilmu Komputer yang tidak dapat penulis sebutkan namanya satu per satu, yang telah menjadi teman yang baik bagi penulis, senantiasa memberikan dukungan terhadap sistem informasi yang dikembangkan serta dukungan moril dan motivasi, sehingga mampu terus menyemangati penulis untuk segera menyelesaikan penelitian yang dikerjakan.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang telah berkontribusi pada proses pengerjaan penelitian ini. Tidak lupa penulis sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, saran dan kritik yang membangun merupakan hal-hal yang penulis harapkan dapat disampaikan kepada penulis.

Malang, 2 Juli 2018



Penulis

danniarreza@gmail.com

ABSTRAK

Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya adalah fakultas dibawah naungan Universitas Brawijaya, yang dalam menyelenggarakan kegiatan perkuliahnya tidak lepas dari penggunaan sarana prasarana. Seiring dengan berjalannya kegiatan tersebut, sarana prasarana akan mengalami kerusakan yang menyebabkan berkurangnya fungsi dan kegunaannya. Solusi sementara adalah dengan civitas akademik FILKOM melaporkan kepada pegawai Perlengkapan langsung secara lisan atau melalui website E-Complaint Universitas Brawijaya. Permasalahan yang muncul dari proses melaporkan keluhan tersebut adalah kesulitan dalam mengidentifikasi sarana prasarana yang dilaporkan sehingga membutuhkan waktu pelaporan yang cukup lama yaitu antara 4 sampai 6 menit.

Berdasarkan permasalahan tersebut, dikembangkanlah Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana yang dapat digunakan civitas akademik untuk melaporkan keluhan sarana prasarana di lingkup wilayah FILKOM UB. Sistem informasi ini juga dapat digunakan untuk melacak status pelaporan dari keluhan-keluhan yang telah dilaporkan, baik oleh pihak pelapor sendiri maupun oleh civitas akademik lainnya. Pengembangan dari sistem informasi ini dibagi menjadi pengembangan aplikasi *android* untuk pelapor dan *website operator* untuk pegawai Perlengkapan. Metode pengembangan perangkat lunak yang digunakan untuk mengembangkan sistem informasi ini adalah menggunakan *waterfall model*. Kemudian untuk menguji hasil implementasi dari fungsi pelaporan dan pelacakan status pelaporan, akan dilakukan pengujian *validation*, pengujian *usability* dan pengujian perbandingan waktu. Dari tahap pengujian didapatkan hasil bahwa dengan disediakannya fitur kamera dan pindai kode QR, sistem informasi yang dikembangkan dapat mempercepat proses pelaporan keluhan dan juga sistem informasi ini dapat mempercepat pelacakan status pelaporan sarana prasarana.

Kata kunci: Sistem Informasi, Pelaporan, Pelacakan, Sarana Prasarana, Android, Website, QR Code

ABSTRACT

Faculty of Computer Science (FILKOM) Brawijaya University is a faculty under Brawijaya University, which in carrying out administrative and academic activities can not be separated from the use of infrastructure facilities. As those activities goes by, infrastuctures will eventually break and causing the reduction of function and its usefulness. Currently applied solution is by the academic community of FILKOM reporting to the Equipment staff spoken directly or through the E-Complaint website of Brawijaya University. The problem that arises from the process of reporting the complaint is the difficulty in identifying the infrastructure being reported so that it takes a long reporting time of between 4 to 6 minutes.

Based on these problems, Infrastructure Reporting Management Information System that can be used by academic community to report complaints of infrastructure facilities in region of FILKOM UB are developed. This information system can also be used to track the status of reported complaints, either by the reporting party itself or by other academic community. The development of this information system is divided into development of android applications for reporters and operators website for Equipment staff. Software development method used to develop this information system is using waterfall model. Then to test the implementation result of the reporting and status tracking function, validation testing, usability testing and time comparison testing were performed. From the testing phase it was found that with the availability of camera features and QR code scan, the developed information system can accelerate the complaint reporting process and also this information system can accelerate the tracking of the reporting status of infrastructure facilities.

Keywords: Information System, Reporting, Tracking, Infrastructure Facility, Android, Website, QR Code.

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	4
1.3 Tujuan	4
1.4 Manfaat.....	5
1.5 Batasan masalah	5
1.6 Sistematika pembahasan.....	5
BAB 2 LANDASAN KEPUSTAKAAN	7
2.1 Kajian Pustaka	7
2.2 Profil Fakultas Ilmu Komputer Universitas Brawijaya	9
2.3 Manajemen Aset.....	10
2.3.1 Sistem Informasi Manajemen Aset.....	11
2.4 Sistem Informasi	11
2.5 Software Development Life Cycle (SDLC)	12
2.6 Waterfall Model.....	12
2.7 Tahap Analisis	13
2.7.1 Analisis Proses Bisnis.....	13
2.7.2 Analisis Kebutuhan Fungsional dan Non-Fungsional	17
2.8 Tahap Perancangan	17
2.8.1 Unified Modelling Language	18
2.8.2 Physical Data Model.....	23

2.9 Tahap Implementasi	24
2.9.1 Framework CodeIgniter	25
2.9.2 Aplikasi <i>Android</i>	26
2.9.3 Web Service.....	27
2.9.4 JSON	27
2.9.5 Quick Response Code (<i>QR Code</i>).....	28
2.10 Tahap Pengujian	30
2.10.1 <i>Black-Box Testing</i>	30
2.10.2 <i>Usability Testing</i>	31
2.11 Tahap Pemeliharaan	34
BAB 3 METODOLOGI	35
3.1 Metodologi Penelitian	35
3.1.1 Wawancara	35
3.1.2 Studi Pustaka.....	36
3.1.3 Analisis Kebutuhan Sistem	36
3.1.4 Perancangan Sistem	37
3.1.5 Implementasi Sistem.....	37
3.1.6 Pengujian Sistem	37
3.1.7 Kesimpulan dan Saran.....	37
BAB 4 ANALISIS KEBUTUHAN	38
4.1 Rekayasa Kebutuhan.....	38
4.1.1 Analisis Proses Bisnis As-Is	38
4.1.2 Elisitasi Kebutuhan	49
4.1.3 Identifikasi Aktor	50
4.1.4 Pemodelan Proses Bisnis <i>To-Be</i>	51
4.1.5 Kebutuhan Fungsional.....	56
4.2 Pemodelan Kebutuhan	64
4.2.1 Pemodelan Use Case Diagram	64
4.2.2 Pemodelan <i>Use Case Scenario</i> Pelapor.....	67
4.2.3 Pemodelan <i>Use Case Scenario Operator</i>	73
BAB 5 PERANCANGAN.....	85
5.1 Perancangan Arsitektur Sistem.....	85

5.2 Perancangan <i>Sequence Diagram</i>	86
5.2.1 Perancangan <i>Sequence Diagram</i> Pelapor	86
5.2.2 Perancangan <i>Sequence Diagram</i> Operator.....	90
5.3 Perancangan <i>Class Diagram</i>	95
5.3.1 Perancangan <i>Class Diagram</i> Pelapor	95
5.3.2 Perancangan <i>Class Diagram</i> Operator	97
5.4 Perancangan Skema Basis Data	100
5.5 Perancangan Antarmuka Pengguna	101
5.5.1 Perancangan Antarmuka Pengguna Pelapor	102
5.5.2 Perancangan Antarmuka Pengguna Operator	104
5.6 Perancangan Algoritme	107
5.6.1 Perancangan Algoritme Pelapor	107
5.6.2 Perancangan Algoritme Operator	108
BAB 6 IMPLEMENTASI	110
6.1 Batasan Implementasi	110
6.2 Implementasi Program	110
6.2.1 Implementasi Aplikasi <i>Mobile</i> Pelapor	110
6.2.2 Implementasi <i>Website Operator</i>	116
6.2.3 Implementasi Web Service	121
6.2.4 Implementasi Data Definition Language (DDL).....	127
6.3 Implementasi Antarmuka	132
6.3.1 Implementasi Antarmuka Aplikasi <i>Mobile</i> Pelapor	132
6.3.2 Implementasi Antarmuka <i>Website Operator</i>	138
BAB 7 PENGUJIAN DAN ANALISIS HASIL	145
7.1 Pengujian <i>Black-Box</i>	145
7.1.1 Pengujian Validasi Aplikasi Pelapor	145
7.1.2 Pengujian Validasi <i>Website Operator</i>	149
7.2 Pengujian <i>Usability</i>	151
7.2.1 Metrik <i>Effectiveness</i>	151
7.2.2 Metrik <i>Efficiency</i>	156
7.2.3 Metrik <i>Satisfaction</i>	159
7.3 Pengujian Perbandingan Waktu	163

7.3.1 Pengujian Waktu Aktifitas To-Be	163
7.3.2 Analisis Perbandingan Waktu	165
BAB 8 KESIMPULAN DAN SARAN	168
8.1 Kesimpulan.....	168
8.2 Saran	169
DAFTAR PUSTAKA.....	171
LAMPIRAN A HASIL WAWANCARA PEGAWAI PERLENGKAPAN.....	174
LAMPIRAN B HASIL WAWANCARA MAHASISWA CIVITAS FILKOM UB	176
LAMPIRAN C HASIL KUESIONER <i>TASK LEVEL SATISFACTION</i> PELAPOR.....	180
LAMPIRAN D HASIL KUESIONER <i>TASK LEVEL SATISFACTION OPERATOR</i>	190



DAFTAR TABEL

Tabel 2.1 Analisis Permasalahan	14
Tabel 2.2 Notasi pada BPMN	14
Tabel 2.3 Simbol <i>Use Case Diagram</i>	19
Tabel 2.4 Simbol <i>Sequence Diagram</i>	21
Tabel 2.5 Penjelasan simbol <i>Class Diagram</i>	22
Tabel 2.6 Bobot Nilai Jawaban	33
Tabel 2.7 Persentase Nilai Jawaban	33
Tabel 4.1 Penyelesaian Waktu Aktifitas Pelaporan As-Is Secara Langsung	41
Tabel 4.2 Penyelesaian Waktu Aktifitas Pelacakan Status Pelaporan As-Is secara lisan dan <i>Instant Messenger</i>	42
Tabel 4.3 Penyelesaian Waktu Aktifitas Pelaporan As-Is melalui <i>E-Complaint</i>	45
Tabel 4.4 Penyelesaian Waktu Aktifitas Pelacakan Status Pelaporan As-Is melalui <i>E-Complaint</i>	47
Tabel 4.5 Analisis Permasalahan.....	48
Tabel 4.6 Temuan Kebutuhan Pihak Pelapor	49
Tabel 4.7 Temuan Kebutuhan Pihak <i>Operator</i>	50
Tabel 4.8 Identifikasi Aktor	50
Tabel 4.9 Daftar Proses Bisnis <i>To-Be</i>	51
Tabel 4.10 Daftar Aktifitas Proses Bisnis PB-SIMA-01	54
Tabel 4.11 Daftar Aktifitas Proses Bisnis PB-SIMA-02	56
Tabel 4.12 Kebutuhan Fungsional Pelapor	56
Tabel 4.13 Kebutuhan Fungsional <i>Operator</i>	59
Tabel 4.14 Use Case Scenario Mendaftarkan Akun Pelapor.....	67
Tabel 4.15 Use Case Scenario Login Pelapor	68
Tabel 4.16 Use Case Scenario Mengirim Laporan.....	68
Tabel 4.17 Use Case Scenario Melihat Laporan Lini Masa.....	69
Tabel 4.18 Use Case Scenario Melihat Laporan yang Sedang Ditangani	70
Tabel 4.19 Use Case Scenario Melihat Laporan Riwayat	71
Tabel 4.20 Use Case Scenario Logout Pelapor	72
Tabel 4.21 Use Case Scenario Mendaftarkan Akun Operator	73

Tabel 4.22 Use Case Scenario Login Operator	73
Tabel 4.23 Use Case Scenario Melihat Laporan	74
Tabel 4.24 Use Case Scenario Menindak Lanjuti Laporan	75
Tabel 4.25 Use Case Scenario Menambah Data Barang	77
Tabel 4.26 Use Case Scenario Memperbarui Data Barang	78
Tabel 4.27 Use Case Scenario Menghapus Data Barang.....	78
Tabel 4.28 Use Case Scenario Menambah Data Teknisi	79
Tabel 4.29 Use Case Scenario Memperbarui Data Teknisi	80
Tabel 4.30 Use Case Scenario Mendisposisi Teknisi	81
Tabel 4.31 Use Case Scenario Menghapus Data Teknisi.....	82
Tabel 4.32 Use Case Scenario Melihat Daftar Pelapor	82
Tabel 4.33 Use Case Scenario Logout Operator.....	83
Tabel 5.1 Perancangan Algoritme Mengirim Laporan	107
Tabel 5.2 Perancangan Algoritme Melihat Laporan	108
Tabel 5.3 Perancangan Algoritme Menerima Laporan	108
Tabel 5.4 Perancangan Algoritme Menindaklanjuti Laporan	109
Tabel 6.1 Implementasi Mengirim Laporan	111
Tabel 6.2 Implementasi Fungsi Kamera dan Pindai QR Code	113
Tabel 6.3 Implementasi Melihat Laporan Lini Masa	114
Tabel 6.4 Implementasi Javascript Notifikasi Menerima Laporan.....	116
Tabel 6.5 Implementasi Controller ReportController Menerima Laporan	117
Tabel 6.6 Implementasi Model Report Menerima Laporan	118
Tabel 6.7 Implementasi Controller ReportController Menindaklanjuti Laporan	118
Tabel 6.8 Implementasi Model Report Menindaklanjuti Laporan.....	120
Tabel 6.9 Implementasi Controller ReportController Mendisposisi Teknisi.....	120
Tabel 6.10 Implementasi Model Report Menindaklanjuti Laporan.....	121
Tabel 6.11 Implementasi Model Technician Menindaklanjuti Laporan	121
Tabel 6.12 Implementasi Controller ReportController Mendaftarkan Akun.....	121
Tabel 6.13 Implementasi Controller ReportController Menambahkan Laporan	123
Tabel 6.14 Implementasi Controller ReportController Mendapatkan Laporan Riwayat.....	124
Tabel 6.15 Implementasi Model Report Mendapatkan Laporan Riwayat.....	124

Tabel 6.16 Implementasi DDL Tabel Asset	127
Tabel 6.17 Implementasi DDL Tabel Assetstatus	128
Tabel 6.18 Implementasi DDL Tabel Assettype	128
Tabel 6.19 Implementasi DDL Tabel Civitas	128
Tabel 6.20 Implementasi DDL Tabel Student.....	129
Tabel 6.21 Implementasi DDL Tabel Lecturer	129
Tabel 6.22 Implementasi DDL Tabel Staff	129
Tabel 6.23 Implementasi DDL Tabel EquipmentStaff	130
Tabel 6.24 Implementasi DDL Tabel Technician	130
Tabel 6.25 Implementasi DDL Tabel TechnicianStatus	131
Tabel 6.26 Implementasi DDL Tabel Report	131
Tabel 6.27 Implementasi DDL Tabel ReportStatus	132
Tabel 6.28 Implementasi DDL Tabel ReportRecord	132
Tabel 7.1 Kasus Uji Pengujian Validasi Mengirim Laporan	146
Tabel 7.2 Kasus Uji Pengujian Validasi Melihat Laporan Lini Masa	147
Tabel 7.3 Kasus Uji Pengujian Validasi Melihat Laporan yang Sedang Ditangani.....	147
Tabel 7.4 Kasus Uji Pengujian Validasi Melihat Laporan Riwayat.....	148
Tabel 7.5 Tabel Hasil Pengujian Validasi Aplikasi Mobile Pelapor	148
Tabel 7.6 Kasus Uji Pengujian Validasi Menerima Laporan	149
Tabel 7.7 Kasus Uji Pengujian Validasi Menindaklanjuti Laporan	150
Tabel 7.8 Tabel Hasil Pengujian Validasi Website Operator	150
Tabel 7.9 Daftar Task Usability Testing Pengguna Pelapor.....	151
Tabel 7.10 Hasil Keberhasilan Task Pengguna Pelapor.....	153
Tabel 7.11 Daftar Task Usability Testing Pengguna Operator	154
Tabel 7.12 Hasil Keberhasilan Task Pengguna Operator	156
Tabel 7.13 Hasil Pengukuran Waktu Task Pengguna Pelapor.....	157
Tabel 7.14 Hasil Pengukuran Waktu Task Pengguna Operator	158
Tabel 7.15 Kuesioner Task Level Satisfaction Pengguna Pelapor	159
Tabel 7.16 Kuesioner Task Level Satisfaction Pengguna Operator.....	161
Tabel 7.17 Tabel Pengujian Waktu Aktifitas Pelaporan To-Be	164
Tabel 7.18 Tabel Pengujian Waktu Aktifitas Pelaporan To-Be	165
Tabel 7.19 Tabel Hasil Perbandingan Waktu	165

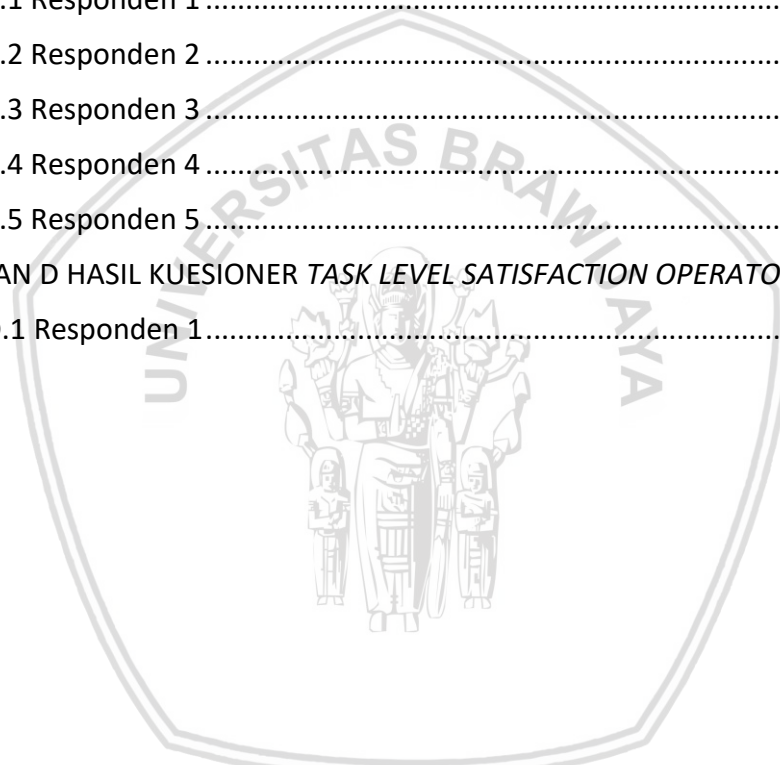
DAFTAR GAMBAR

Gambar 2.1 Susunan Organisasi FILKOM.....	9
Gambar 2.2 Diagram SDLC <i>Waterfall Model</i>	13
Gambar 2.3 Contoh use case diagram	18
Gambar 2.4 Class diagram dari kelas Thoroughbred	22
Gambar 2.5 Contoh <i>Physical Data Model</i>	24
Gambar 2.6 Struktur Kerja CodeIgniter	25
Gambar 2.7 Contoh representasi objek JSON.....	28
Gambar 2.8 Contoh QR Code	29
Gambar 2.9 Contoh Skala Likert.....	33
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	35
Gambar 4.1 Proses Bisnis <i>As-Is</i> Pelaporan Keluhan Sarana Prasarana Secara Langsung	39
Gambar 4.2 Proses Bisnis <i>As-Is</i> Pelaporan Keluhan Sarana Prasarana Melalui E-Complaint UB	43
Gambar 4.3 Proses Bisnis To-Be PB-SIMA-01	52
Gambar 4.4 Proses Bisnis <i>To-Be</i> PB-SIMA-02.....	54
Gambar 4.5 <i>Use Case Diagram</i> Pelapor Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.....	65
Gambar 4.6 <i>Use Case Diagram Operator</i> Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.....	66
Gambar 5.1 <i>Architecture Diagram</i> Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.....	85
Gambar 5.2 Sequence Diagram Mengirim Laporan.....	87
Gambar 5.3 Sequence Diagram Melihat Laporan Lini Masa.....	88
Gambar 5.4 Sequence Diagram Melihat Laporan yang Sedang Ditangani	89
Gambar 5.5 Sequence Diagram Melihat Laporan Riwayat	90
Gambar 5.6 Sequence Diagram Menerima Laporan	91
Gambar 5.7 Sequence Diagram Menindaklanjuti Laporan.....	92
Gambar 5.8 Sequence Diagram Mendisposisi Teknisi	95
Gambar 5.9 Class Diagram Aplikasi Mobile Pelapor	96
Gambar 5.10 Class Diagram Relasi Antar Kelas Website Operator	97

Gambar 5.11 Class Diagram Relasi Antar Kelas Controller	98
Gambar 5.12 Class Diagram Relasi Antar Kelas Model	99
Gambar 5.13 Contoh penggambaran physical data model	100
Gambar 5.14 Physical Data Model Basis Data SIMA.....	101
Gambar 5.15 Rancangan Halaman Login dan Sign Up Pelapor	102
Gambar 5.16 Rancangan Halaman Laporan Baru	103
Gambar 5.17 Rancangan Halaman Lini Masa, yang Sedang Ditnagani dan Laporan Riwayat.....	103
Gambar 5.18 Rancangan Halaman Rincian Laporan	104
Gambar 5.19 Rancangan Halaman Login dan Sign Up Operator	104
Gambar 5.20 Rancangan Halaman Daftar Laporan, Aset, Teknisi dan Pelapor..	105
Gambar 5.21 Rancangan Halaman Rincian Laporan, Aset, Teknisi dan Pelapor	106
Gambar 5.22 Rancangan Halaman Disposisi Teknisi	106
Gambar 6.1 Format JSON Mendapatkan Laporan Riwayat	127
Gambar 6.2 Implementasi Halaman Login Pelapor	133
Gambar 6.3 Implementasi Halaman Sign Up Pelapor.....	134
Gambar 6.4 Implementasi Halaman Laporan Baru Inventaris.....	135
Gambar 6.5 Implementasi Halaman Laporan Lini Masa	136
Gambar 6.6 Implementasi Halaman Rincian Laporan	137
Gambar 6.7 Implementasi Halaman Login Operator	139
Gambar 6.8 Implementasi Halaman Sign Up Operator	139
Gambar 6.9 Implementasi Halaman Daftar Laporan dan Notifikasi Laporan.....	140
Gambar 6.10 Implementasi Halaman Daftar Aset	141
Gambar 6.11 Implementasi Halaman Rincian Laporan	142
Gambar 6.12 Implementasi Halaman Rincian Aset	143
Gambar 6.13 Implementasi Halaman Disposisi Teknisi	144
Gambar 7.1 Aturan penomoran pengujian validasi pelapor	145
Gambar 7.2 Aturan penomoran pengujian validasi operator.....	149
Gambar 7.3 Grafik Batang Hasil Tingkat Penyelesaian Pengguna Pelapor.....	154
Gambar 7.4 Grafik Batang Tingkat TBE Pengguna Pelapor.....	158
Gambar 7.5 Grafik Batang Perbandingan Waktu Pelaporan	166
Gambar 7.6 Grafik Batang Perbandingan Waktu Pelacakan Status	167

DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA PEGAWAI PERLENGKAPAN	174
A.1 Wawancara Pegawai Perlengkapan	174
LAMPIRAN B HASIL WAWANCARA MAHASISWA CIVITAS FILKOM UB	176
B.1 Wawancara Mahasiswa 1	176
B.2 Wawancara Mahasiswa 2	178
LAMPIRAN C HASIL KUESIONER <i>TASK LEVEL SATISFACTION</i> PELAPOR	180
C.1 Responden 1	180
C.2 Responden 2	182
C.3 Responden 3	184
C.4 Responden 4	186
C.5 Responden 5	188
LAMPIRAN D HASIL KUESIONER <i>TASK LEVEL SATISFACTION</i> OPERATOR	190
D.1 Responden 1	190



BAB 1 PENDAHULUAN

1.1 Latar belakang

Teknologi merupakan suatu sarana hasil karya manusia yang sudah menjadi bagian dari kehidupan masyarakat sehari-hari di era modern ini. Salah satu teknologi yang makin berkembang saat ini adalah adanya Sistem Informasi yang dapat berfungsi untuk mengumpulkan data, memprosesnya menjadi informasi dan mendistribusikannya kepada para pengguna (Hall, 2008). Perkembangan dari Sistem Informasi sendiri sudah banyak diterapkan dan banyak dirasakan manfaatnya oleh masyarakat di berbagai instansi, salah satunya adalah di instansi pendidikan perkuliahan yaitu Universitas Brawijaya. Contoh Sistem Informasi yang telah diterapkan oleh Universitas Brawijaya saat ini adalah Sistem Informasi Akademik Mahasiswa (SIAM) dan Sistem Informasi Akademik Dosen (SIADO).

Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya merupakan sebuah fakultas yang berada dibawah naungan Universitas Brawijaya yang dimana dalam menyelenggarakan kegiatan administrasi dan akademik perkuliahan pasti tidak akan luput dari penggunaan fasilitas sarana prasarana. Seiring dengan berjalannya kegiatan administrasi dan akademik perkuliahan, sarana prasarana itu suatu saat akan mengalami penurunan kualitas dan kerusakan yang menyebabkan berkurangnya fungsi, kegunaan dan kinerjanya serta meningkatnya biaya operasional dibandingkan dengan tujuan awalnya (Cagle, 2003). Hal ini dapat mengganggu kegiatan administrasi dan akademik perkuliahan dalam menjalankan fungsinya saat menggunakan sarana prasarana tersebut. Untuk mencegah terjadinya gangguan dan hambatan tersebut, maka diperlukan proses pemeliharaan secara berkala. Melalui pemeliharaan secara berkala, suatu organisasi bisa mengurangi resiko kerusakan asset, mengurangi biaya perbaikan asset, meningkatkan kinerja dari asset yang dimiliki serta meningkatkan kepuasan pengguna dari asset tersebut (Justin, 2012).

Dalam mendukung proses pemeliharaan sarana prasarana di FILKOM sendiri, Pegawai Sub Bagian Umum dan Perlengkapan bertanggung jawab untuk menerima dan menanggapi keluhan mengenai sarana prasarana yang disampaikan oleh seluruh civitas akademik yang ada di FILKOM. Sarana prasarana yang keluhannya harus ditanggapi oleh Kasubag Umum dan Perlengkapan pun jenisnya bermacam-macam dan tersebar diseluruh lingkup area FILKOM, baik dari perlengkapan inventaris kantor untuk mendukung kinerja pegawai maupun perlengkapan penunjang akademik perkuliahan di masing-masing ruang kuliah. Penyampaian keluhan mengenai sarana prasarana sendiri selama ini bisa dilakukan dengan berbagai cara, baik secara lisan langsung bertemu pihak Perlengkapan, melalui media komunikasi seperti telepon, *sms*, *instant messaging*, maupun melalui *E-Complaint* Universitas Brawijaya.

Permasalahan muncul ketika ada keluhan mengenai salah satu sarana prasarana yang disampaikan dari civitas akademik FILKOM, kemudian beberapa saat setelahnya terdapat keluhan yang sama mengenai sarana prasarana yang telah dilaporkan sebelumnya oleh pelapor lain ketika sebenarnya Kasubag Umum dan Perlengkapan sudah menerima dan menanggapi keluhan oleh pelapor pertama tersebut. Pihak pelapor beserta seluruh civitas akademik FILKOM lainnya sering kali tidak mendapat informasi status tanggapan dari pihak Kasubag Umum dan Perlengkapan, bahkan setelah keluhan mengenai sarana prasarana tersebut sudah diterima atau sedang ditindak lanjuti. Hal ini mengakibatkan pihak pelapor mengalami kesulitan untuk melakukan pelacakan status mengenai keluhan yang telah disampaikan sebelumnya.

Dalam menyampaikan keluhan mengenai segala hal yang berhubungan dengan kegiatan akademik dan administratif termasuk mengenai sarana prasarana, Universitas Brawijaya telah menyediakan layanan *E-Complaint* berbasis *website* yang dapat diakses di <https://e-complaint.ub.ac.id/>. Menurut pengakuan dari pihak pelapor khususnya dari mahasiswa, dalam melaporkan keluhan mengenai sarana prasarana pada layanan *E-Complaint* tersebut dirasa kurang mudah untuk mengidentifikasi suatu barang yang akan dilaporkan. Pelapor diharuskan untuk menjelaskan ciri-ciri dari sarana prasarana yang dilaporkan sedetail mungkin agar proses identifikasi sarana prasarana dapat dilakukan. Proses input laporan keluhan untuk menjelaskan subjek laporan, deskripsi permasalahan sekaligus penjelasan ciri-ciri sarana prasarana yang bermasalah serta usulan solusi dari permasalahan ini bagi pihak pelapor membutuhkan waktu antara 4-6 menit. Setelah laporan keluhan tersebut diterima oleh pegawai Perlengkapan, pegawai yang melakukan survey juga masih harus mencari sarana prasarana yang bermasalah hanya berdasarkan penjelasan teks yang diterima.

Selain itu berdasarkan pengakuan dari Ferix Panji Andrianto selaku pegawai Perlengkapan, proses masuknya keluhan dari *E-Complaint* ke pegawai Perlengkapan selama ini membutuhkan waktu paling cepat 4 hari dan paling lama 10 hari kerja. Satu hari kerja yang berlaku di Universitas Brawijaya terdiri dari 8 jam kerja, maka proses masuknya keluhan dari *E-Complaint* ke pegawai Perlengkapan dapat membutuhkan waktu paling cepat selama 32 jam dan paling lama 80 jam kerja. Hal ini disebabkan karena *administrator* dari sistem *E-Complaint* UB untuk FILKOM dikelola oleh Sekretaris FILKOM, sehingga segala jenis laporan keluhan dari *E-Complaint* UB harus melalui *administrator* Sekretaris tersebut terlebih dahulu. Setelah laporan keluhan tersebut telah benar diidentifikasi sebagai keluhan terhadap sarana prasarana, maka kemudian oleh Sekretaris akan dibuatkan surat keluhan dan disampaikan kepada pegawai Perlengkapan untuk ditindaklanjuti. Padahal selama suatu sarana prasarana mengalami kegagalan dan tidak segera ditangani, selain kinerja dari sarana prasarana tersebut menjadi tidak optimal, biaya operasional dari layanan pengguna sarana prasarana tersebut juga mengalami peningkatan (Cagle, 2003). Selain itu, salah satu tujuan utama dari manaj (Cagle, 2003)emen aset sendiri adalah agar suatu organisasi dapat memantau dan menyelesaikan permasalahan mengenai aset yang dimilikinya secara *real-time* (Mehta & Reddy, 2014).

Masalah mengenai sarana prasarana di FILKOM Universitas Brawijaya merupakan permasalahan yang membahas mengenai Manajemen Aset. Manajemen Aset sendiri merupakan kegiatan pengelolaan aset fisik melalui pendekatan yang sistematis (Hastings, 2010). Barang-barang sarana prasarana yang dibahas pada penelitian ini dapat disebut sebagai aset karena aset merupakan barang-barang fisik seperti barang perlengkapan, kendaraan, mesin, bangunan, pabrik, hingga informasi terkait kendali teknis serta sistem perangkat lunak yang digunakan untuk menjalankan fungsi bisnis organisasi (Hastings, 2010). Manajemen Aset diperlukan dalam suatu organisasi karena memiliki kemampuan untuk memberikan pengetahuan mengenai aset dan kapasitasnya untuk kegiatan manajemen dan aktivitas pendukung keputusan yang terkait dengan konteks bisnis organisasi tersebut. Dalam rangka membantu pelaku yang terlibat untuk mendokumentasikan fungsi dari Manajemen Aset, maka dibutuhkan suatu Sistem Informasi Manajemen Pelaporan yang berbasis computer.

Dalam melakukan serangkaian aktivitas yang berkaitan dengan mengembangkan sebuah perangkat lunak, diperlukan pengetahuan dan penerapan mengenai empat aktivitas proses fundamental (Sommerville, 2007). Empat aktivitas proses tersebut meliputi proses *Software Specification*, *Software Development*, *Software Validation and Verification*, *Software Evolution*. Rangkaian aktivitas proses ini dikenal dengan istilah *Software Development Life Cycle* (SDLC). *Software Specification* merupakan tahap untuk menentukan spesifikasi kebutuhan dan batasan-batasan dari perangkat lunak yang akan dibuat. *Software Development* adalah tahap untuk merancang dan membuat perangkat lunak itu sendiri. Setelah itu dilanjutkan ke tahap *Software Validation and Verification*, yaitu tahap untuk menguji dan memastikan bahwa perangkat lunak yang dibuat telah sesuai dengan perancangan dan kebutuhan pengguna di awal. Sedangkan *Software Evolution* adalah ketika perangkat lunak yang telah dibuat dimodifikasi untuk menyesuaikan terhadap perubahan kebutuhan pengguna. Dalam penelitian ini nantinya aktivitas proses fundamental yang akan dilakukan hanya terbatas sampai *Software Validation* saja karena perubahan kebutuhan pengguna tidak akan dilibatkan.

Maka berdasarkan latar belakang yang telah dijabarkan, peneliti tertarik untuk melakukan penelitian dengan judul **“PENGEMBANGAN SISTEM INFORMASI MANAJEMEN PELAPORAN SARANA DAN PRASARANA STUDI PADA FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA”**. Hasil yang didapatkan dari penelitian ini diharapkan bisa digunakan oleh Fakultas Ilmu Komputer Universitas Brawijaya sebagai acuan dalam mengembangkan sistem informasi yang mampu membantu seluruh civitas akademik FILKOM dalam melakukan pelaporan mengenai sarana prasarana serta membantu Pegawai Perlengkapan dalam melakukan penanganan laporan dan pemeliharaan sarana prasarana tersebut.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah yang dapat diambil antara lain:

1. Bagaimanakah hasil evaluasi proses bisnis pelaporan keluhan dan pelacakan status pelaporan keluhan sarana prasarana yang sedang berlaku di Fakultas Ilmu Komputer Universitas Brawijaya saat ini?
2. Bagaimanakah hasil analisis kebutuhan Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil evaluasi proses bisnis?
3. Bagaimanakah hasil perancangan Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil analisis kebutuhan sistem?
4. Bagaimana hasil implementasi Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil perancangan sistem?
5. Bagaimana hasil pengujian Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil implementasi sistem?

1.3 Tujuan

Berdasarkan masalah yang telah dirumuskan sebelumnya, maka tujuan dari penulisan skripsi ini antara lain adalah:

1. Untuk mengetahui hasil evaluasi dari proses bisnis pelaporan keluhan dan pelacakan status pelaporan keluhan sarana prasarana yang sedang berlaku di Fakultas Ilmu Komputer Universitas Brawijaya saat ini.
2. Untuk mengetahui hasil dari analisis kebutuhan Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil evaluasi proses bisnis.
3. Untuk mengetahui hasil dari perancangan Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil analisis kebutuhan sistem.
4. Untuk mengetahui hasil dari implementasi Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil perancangan sistem.
5. Untuk mengetahui hasil dari pengujian Sistem Informasi Pelaporan dan Pemeliharaan Sarana Prasarana di Fakultas Ilmu Komputer Universitas Brawijaya berdasarkan hasil implementasi sistem.

1.4 Manfaat

Penelitian ini diharapkan memberi manfaat baik bagi instansi maupun bagi peneliti, antara lain:

1. Menghasilkan sistem berupa aplikasi perangkat bergerak yang dapat digunakan oleh civitas akademik FILKOM untuk melaporkan keluhan sarana prasarana di sekitar lingkup wilayah FILKOM.
2. Menghasilkan sistem berbasis *website* yang dapat digunakan oleh pegawai Perlengkapan untuk menerima dan menangani laporan keluhan mengenai sarana prasarana agar dapat segera dilakukan pemeliharaan sarana prasarana yang terlapor.
3. Memperbaiki sistem lama sehingga mampu menambah efisiensi kerja bagi Pegawai Umum dan Perlengkapan dalam melakukan pemeliharaan sarana prasarana.

1.5 Batasan masalah

Berdasarkan latar belakang yang telah dijabarkan sebelumnya, maka batasan masalah untuk membatasi permasalahan agar tidak melebar melebihi pokok permasalahan antara lain adalah:

1. Jenis aset yang dibahas hanya sebatas aset sarana prasarana yang dimiliki oleh FILKOM Universitas Brawijaya.
2. Pengembangan terbatas pada aktifitas pelaporan dan pelacakan keluhan sarana prasarana oleh civitas akademik FILKOM yang diterima dan ditangani oleh pegawai Pegawai Perlengkapan FILKOM Universitas Brawijaya.

1.6 Sistematika pembahasan

Keseluruhan dari penulisan penelitian ini dibagi menjadi delapan bab yang dibagi tiap pokok bahasan. Adapun sistematika dari penelitian ini sebagai berikut:

Bab I : Pendahuluan

Berisi tentang latar belakang penelitian, rumusan masalah, tujuan penulisan, manfaat penulisan, dan batasan masalah yang akan diselesaikan.

Bab II : Landasan Kepustakaan

Berisi tentang dasar teori dan referensi yang mendasari penelitian ini.

Bab III : Metodologi

Berisi tentang penjelasan langkah-langkah/metodologi dalam melakukan penelitian.

Bab IV : Analisis Kebutuhan

Berisi tentang kebutuhan dari sistem yang akan dibuat dalam penelitian ini.

Bab V : Perancangan

Berisi tentang rancangan sistem yang dibuat berdasarkan analisis kebutuhan yang dijabarkan pada bab sebelumnya.

Bab VI : Implementasi

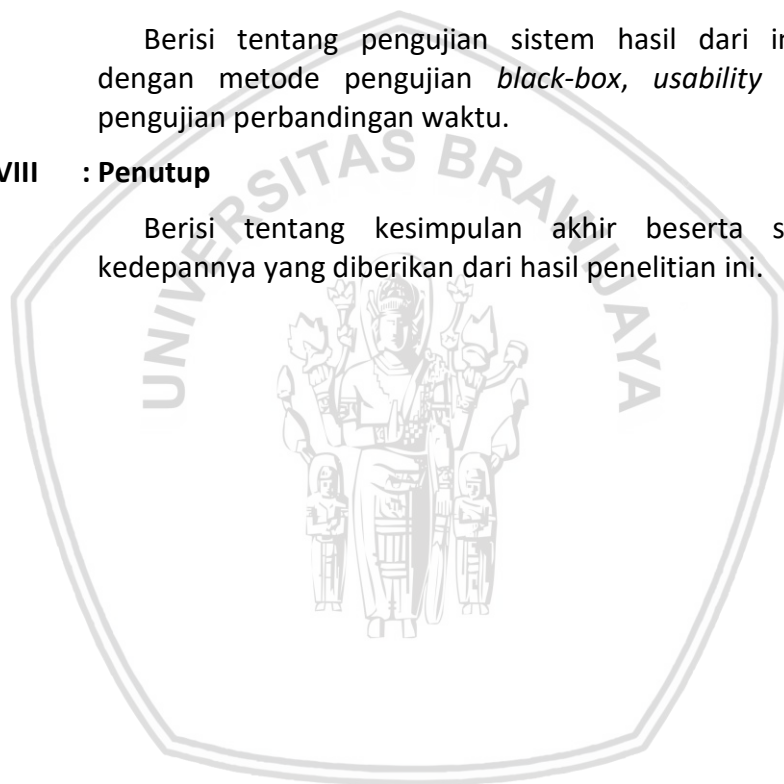
Berisi tentang hasil implementasi dari sistem yang dibuat dalam penelitian ini.

Bab VII : Pengujian dan Analisis Hasil

Berisi tentang pengujian sistem hasil dari implementasi dengan metode pengujian *black-box*, *usability testing* dan pengujian perbandingan waktu.

Bab VIII : Penutup

Berisi tentang kesimpulan akhir beserta saran untuk kedepannya yang diberikan dari hasil penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Penelitian pertama yang dijadikan kajian pustaka pada penelitian dan penulisan skripsi ini antara lain adalah penelitian yang dilakukan oleh Elvi Fetrina, Eri Rustamaji, Tatat Nuraeni dan Yusuf Durrachman (2017) dengan judul *"Inventory Management Information System Development At BPRTIK Kemkominfo Jakarta"*. Penelitian tersebut berfokus pada pengembangan sistem informasi berbasis *website* dan basis data terpusat untuk memfasilitasi kinerja dari divisi pengelolaan kepemilikan aset dan properti negara, mulai dari proses kendali, perawatan, pengisian, pembelanjaan, layanan eksternal, penerimaan sampai dengan penugasan terhadap proses pelaporan. Penelitian tersebut menjelaskan bahwa kegiatan pengelolaan aset yang tidak diatur dengan baik, seperti dengan masih digunakannya laporan yang masih disimpan dalam bentuk *hardcopy* dan tidak disimpan dalam satu basis data yang terpusat, akan rentan terhadap duplikasi dan data yang tidak tersinkronisasi. Proses penugasan dan pemantauan juga masih dilakukan secara manual menggunakan memo bahkan terkadang juga sebatas secara lisan. Hal ini dapat berdampak kepada redundansi data dan kesulitan dalam pemantauan dan pengawasan hasil dari penugasan yang telah diberikan. Saran yang dijelaskan pada penelitian tersebut adalah sistem yang telah dikembangkan dapat ditingkatkan kegunaannya apabila dilakukan pengembangan lebih lanjut dari sistem dalam bentuk aplikasi perangkat bergerak.

Penelitian tentang Pengembangan Sistem Informasi Manajemen Asset Transportasi dan Sistem Pengumpulan Data menggunakan aplikasi perangkat bergerak telah dilakukan sebelumnya oleh Ghazan Khan, Andreas Bueff, Ivan Mihov, Nati Tessema, Javier Garrido, Chris Russel dan Arash Parnia (2016) dengan judul *"Development of Transportation Asset Management and Data Collection System (TAMS) using Mobile Application"*. Penelitian tersebut berfokus pada pengembangan sistem informasi yang memanfaatkan perangkat bergerak untuk mengumpulkan, mengelola, dan memperbarui data aset transportasi di lapangan. Selain itu, *TAMS* memiliki sistem *back-end* yang dikembangkan untuk dapat mengelola akun pengguna, memelihara keamanan dan integritas dari proses pengumpulan data, menyediakan kemampuan untuk mensinkronisasi pengumpulan data dari beberapa pengguna, serta mengeluarkan data ke berbagai macam *format* untuk penggunaan diluar *TAMS*. Sistem *front-end* dari *TAMS* dikembangkan menggunakan bahasa *Java* untuk platform *Android*, kemudian untuk platform *iOS* dikembangkan menggunakan bahasa *Objective-C* dan *Swift*. Sedangkan untuk antarmuka web dan sistem *back-end* dari *TAMS* menggunakan beberapa antarmuka dan bahasa pemrograman yang berbeda seperti *Django*, *Python*, *PHP* dan beberapa *API* lainnya yang mendukung. Metode yang digunakan oleh pengembang untuk pengembangan perangkat lunak *TAMS* ini adalah menggunakan metode *Agile*. Pengembangan dari *TAMS* ini menyediakan sebuah sistem yang mampu menghemat biaya secara efektif dan efisien kepada agen transportasi profesional agar dapat selalu mendapatkan, mengumpulkan dan

memperbarui data mengenai aset transportasi yang sedang berjalan pada basis data yang telah ada.

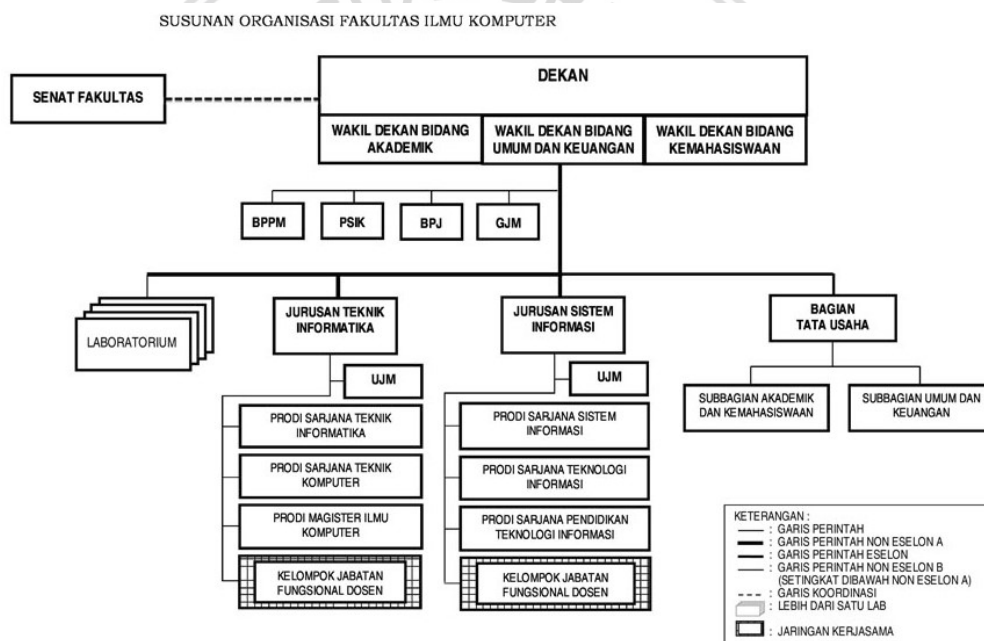
Penelitian ketiga yang dijadikan sebagai kajian pustaka mengenai Pengembangan Sistem Informasi Manajemen Aset menggunakan aplikasi *mobile* adalah penelitian yang dilakukan oleh Igor Bisio, Andrea Sciarrone dan Sandro Zappatore (2016) dengan judul “*A New Asset Tracking Architecture Integrating RFID, Bluetooth Low Energy Tags and Ad Hoc Smartphone Applications*”. Penelitian ini membahas mengenai masalah yang terjadi di situs konstruksi bangunan yang sering kali meliputi lahan kerja yang luas. Masalah yang terjadi adalah di lahan kerja yang luas, efisiensi dari operasi pembangunan dan manufaktur dapat sangat terpengaruh oleh waktu yang dihabiskan untuk mencari lokasi dari aset-aset yang bernilai. Berdasarkan permasalahan tersebut maka solusinya adalah dengan membangun sistem yang dapat melacak lokasi dan pergerakan dari aset fisik. Sistem tersebut melibatkan beberapa mekanisme untuk *tagging* dan *tracking*, yaitu antara lain *Radio Frequency Identification*(RFID) dan *Bluetooth Low Energy*(BLE). Kemudian informasi mengenai *tags* dan aset yang ditandainya disimpan didalam Asset Management *Database System* (AMDS). Tujuan dari penelitian ini adalah agar waktu yang sebelumnya dihabiskan untuk mencari aset yang mengalami salah penempatan, dapat lebih banyak digunakan untuk melakukan optimasi pengambilan keputusan selama kegiatan manufaktur dan konstruksi sehingga efisiensi kerja dan daya tanggap dapat meningkat. Cara kerja dari sistem ini untuk menyimpan lokasi dari aset adalah dengan menempelkan aset-aset dengan penanda RFID, kemudian memasang pembaca RFID di tiap lokasi-lokasi dari situs sehingga setiap periode tertentu pembaca RFID dapat memindai lokasi dari tiap aset dan mengirimkan informasi tersebut ke basis data pusat. Kemudian cara kerja dari fungsi untuk melacak lokasi dari aset adalah dengan menghitung jarak antara *smartphone* dengan *tags* milik suatu aset, mendapatkan lokasi GPS dari *smartphone* lalu mengirim data ke basis data pusat.

Pada Sistem Informasi Manajemen Aset yang akan dikembangkan pada Fakultas Ilmu Komputer memiliki perbedaan pada contoh kasus, bidang manajemen, proses bisnis, metode SDLC dan *platform* pengembangan. Perbedaan contoh kasus dan proses bisnis yang telah disebutkan antara lain adalah sistem informasi ini tidak dikembangkan untuk mengumpulkan dan mengelola data mengenai manajemen aset transportasi, tetapi untuk mengumpulkan dan mengelola data mengenai manajemen aset sarana prasarana pendukung berjalannya kegiatan akademik yang sedang berlangsung di Fakultas Ilmu Komputer (FILKOM). Dengan adanya Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana ini diharapkan dapat membantu seluruh civitas akademika Fakultas Ilmu Komputer untuk secara aktif memantau dan melaporkan kerusakan sarana prasarana yang ada di area cakupan FILKOM. Perbedaan lain yang telah disebutkan sebelumnya adalah pada *platform* pengembangan, karena pada pengembangan Sistem Informasi Manajemen Pelaporan ini akan terbatas pada aplikasi perangkat bergerak pada *platform Android* yang menggunakan bahasa pemrograman *Java* untuk seluruh pengguna civitas akademika sebagai pelapor serta *platform Web Apps* untuk pengguna Pegawai Umum dan Perlengkapan

sebagai pihak yang bertanggung jawab untuk menerima dan menindaklanjuti laporan yang masuk akan menggunakan bahasa pemrograman HTML, PHP, *Javascript* dan beberapa *API* lain yang mendukung. Perbedaan terakhir adalah pada penelitian ini metode pengembangan perangkat lunak yang digunakan adalah menggunakan metode *Waterfall Model* karena sudah adanya proses bisnis mengenai pelaporan dan pemeliharaan yang sudah ditetapkan oleh Kasubag Umum dan Perlengkapan.

2.2 Profil Fakultas Ilmu Komputer Universitas Brawijaya

Fakultas Ilmu Komputer (FILKOM) merupakan salah satu fakultas di Universitas Brawijaya yang berdiri pada tahun 2011 berlokasi di Jl. Veteran Malang, Jawa Timur, 85 Km dari kota Surabaya. FILKOM sebelum tahun 2015 memiliki nama yaitu Program Teknologi Informasi dan Ilmu Komputer atau biasa disingkat PTIIK. FILKOM mempunyai tiga program studi yaitu Program Studi (S-1) Informatika/Ilmu Komputer, Program Studi (S-1) Sistem Komputer, dan Program Studi (S-1) Sistem Informasi.



Gambar 2.1 Susunan Organisasi FILKOM

Sumber : Filkom(2018)

Dalam menjalankan kegiatan administratifnya, FILKOM memiliki susunan organisasi yang terdiri dari beberapa bagian-bagian satuan kerja. Susunan Organisasi dari FILKOM yang dimaksud ditunjukkan sebagaimana pada Gambar 2.1. Pelayanan serta penanganan keluhan mengenai perlengkapan dan sarana prasarana merupakan tanggung jawab dari pegawai yang bekerja dan ditugaskan sebagai Pegawai Perlengkapan. Pegawai Perlengkapan sendiri termasuk ke dalam satuan kerja didalam Sub Bagian Umum dan Perlengkapan, yang mana sub bagian tersebut berada di bawah Bagian Tata Usaha dan berada di bawah garis perintah Wakil Dekan Bidang Umum dan Keuangan.

2.3 Manajemen Aset

Aset merupakan barang fisik seperti pabrik, mesin, bangunan, kendaraan, informasi terkait, kontrol teknis, dan sistem perangkat lunak yang digunakan untuk melayani suatu fungsi bisnis atau organisasi (Hastings, 2010). Pengertian lain dari Aset adalah suatu barang yang mempunyai nilai ekonomi, nilai komersial, atau nilai tukar yang dimiliki oleh badan usaha, instansi atau individu. Aset yang dapat diidentifikasi dalam suatu organisasi apabila dilihat dari bentuknya terdapat beberapa jenis, antara lain adalah:

1. Aset Fisik (*Physical Assets*)

Contoh dari aset jenis ini adalah: Bangunan, mesin, kendaraan, peralatan, infrastruktur.

2. Aset Finansial (*Financial Assets*)

Contoh dari aset jenis ini adalah: Deposito, Obligasi, Saham.

3. Aset Manusia (*Human Assets*)

Contoh dari aset jenis ini adalah: Pegawai, Manajemen, Pelanggan.

4. Aset Informasi (*Information Assets*)

Contoh dari aset jenis ini adalah: Buku Besar Pendanaan, Buku Besar Kode Organisasi.

5. Aset Tidak Berwujud (intangible)

Contoh dari aset jenis ini adalah: Sistem organisasi, tujuan, visi, misi, hak cipta, kualitas, nama baik, budaya, perjanjian dll.

Dilihat dari sudut pandang tujuan bisnis dan organisasi, Manajemen Aset merupakan serangkaian kegiatan yang berhubungan dengan identifikasi kebutuhan aset dan pendanaan, perolehan dan pembaharuan aset serta penyediaan sistem pendukung pemeliharaan aset agar dapat memenuhi tujuan yang diinginkan secara efektif dan efisien (Hastings, 2010). Berdasarkan definisi yang telah disebutkan, maka dapat disimpulkan bahwa Manajemen Aset adalah suatu proses perencanaan, pengadaan, pengelolaan dan perawatan hingga penghapusan suatu sumber daya yang dimiliki oleh individu atau organisasi secara efektif dan efisien dalam rangka mencapai suatu tujuan organisasi. Pengertian lain dari Manajemen Aset adalah proses yang merujuk kepada pengelolaan aset infrastruktur secara proaktif yang terdiri dari pencatatan sistematis dari masing-masing aset; mempertahankan keutuhan aset melalui pemeliharaan, perbaikan, dan penggantian secara terencana serta mengimplementasikan dan mengelola sistem informasi untuk mendukung elemen-elemen yang tersebut (Cagle, 2003).

Dasar pemikiran dalam menerapkan manajemen aset yang proaktif adalah diperlukannya intervensi pada titik-titik strategis selama siklus hidup normal dari suatu aset untuk memperpanjang masa kerja yang diharapkan (Cagle, 2003). Intervensi disini sendiri merupakan langkah turun tangan untuk perbaikan, perawatan prediktif atau preventif, serta untuk pemulihan aset. Langkah ini dalam

artian lain merupakan kegiatan mengeluarkan sumber daya untuk meningkatkan kondisi fisik dan meningkatkan kinerja dengan tujuan untuk memperpanjang masa pelayanan. Semakin panjang masa pelayanan yang bisa diperpanjang sebelum aset tersebut harus benar-benar digantikan, maka *performa* secara keseluruhan juga semakin ekonomis.

2.3.1 Sistem Informasi Manajemen Aset

Sistem Informasi Manajemen Aset adalah sebuah sistem berbasis komputer yang mana dirancang untuk membantu pengguna untuk membuat dan menyimpan dokumentasi mengenai fungsi-fungsi dari manajemen aset (Hastings, 2010). Karena banyaknya sistem seperti ini berasal dari aplikasi di bidang pemeliharaan, maka sistem seperti ini terkadang juga disebut sebagai *Computerized Maintenance Management Systems (CMMS)*. Sistem manajemen yang sudah terkomputerisasi seperti ini sangat penting untuk menangani banyaknya informasi yang masuk terutama yang terkait dengan masalah pengelolaan aset. Hal ini dapat berarti bahwa semakin meluasnya penggunaan suatu sistem komputer, maka semakin besar pula nilai dan manfaat yang akan didapatkan dari sistem tersebut.

Daftar aplikasi di bidang pemeliharaan aset perlu mencakup beberapa fungsi-fungsi yang mendukung aktifitas Manajemen Aset. Beberapa dari fungsi-fungsi yang harus didukung adalah aplikasi tersebut harus mampu menampilkan daftar aset yang dimiliki termasuk daftar aset yang dapat dipelihara. Lalu aplikasi tersebut dapat mendukung fungsi manajemen persediaan *spare part* dan *consumables* untuk keperluan aset yang dimiliki. Kemudian aplikasi tersebut juga harus dapat melakukan *Data Entry* dan *Work Order Management*. Dalam mendukung fungsi dari *Work Order Management*, aplikasi harus dapat menampilkan daftar tugas yang sedang tersedia, pemberitahuan status terkini dari tugas yang sedang tersedia, serta memiliki system pengiriman pesan yang sederhana, cepat, aman dan handal.

2.4 Sistem Informasi

Sistem merupakan sekumpulan elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu (Jogiyanto, 2008). Sedangkan informasi merupakan data yang telah diolah menjadi bentuk yang lebih memiliki arti bagi yang menerimanya (Jogiyanto, 2008). Pengertian sistem Informasi menurut Hall (2008), "sistem informasi adalah serangkaian prosedur *formal* dimana data dikumpulkan, diproses menjadi informasi dan didistribusikan kepada para pengguna". Dari pengertian yang telah dijelaskan diatas, maka dapat disimpulkan bahwa sistem informasi merupakan kumpulan elemen dan prosedur yang memiliki kemampuan untuk mengumpulkan data, mengolah data menjadi informasi serta menyebarkan informasi tersebut kepada pengguna untuk mencapai tujuan atau sasaran tertentu.

Dalam membangun dan mengembangkan sebuah sistem informasi yang mampu untuk memenuhi kebutuhan pengguna dan mencapai tujuan atau sasaran tertentu dari sebuah organisasi, bagi pengembang dari sistem tersebut diperlukan pengetahuan mengenai orang-orang yang akan terlibat dalam penggunaan sistem serta pengetahuan mengenai ruang lingkup seperti apa sistem tersebut nantinya akan digunakan. Selain itu, juga dibutuhkan pengetahuan mengenai proses daur hidup dari pengembangan sistem yang bertujuan untuk menguraikan, menyusun, dan memantau perkembangan dari pengembangan sistem disepanjang tahapan proses tersebut. Proses daur hidup yang dibutuhkan untuk pengembangan sistem tersebut, dapat disebut sebagai proses *Software Development Life Cycle*.

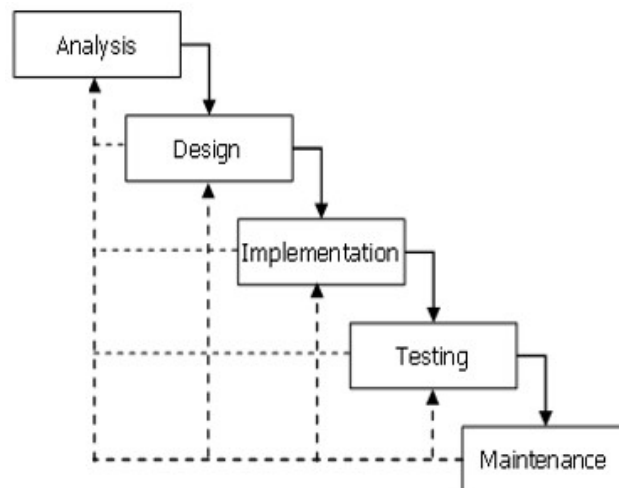
2.5 Software Development Life Cycle (SDLC)

Software Development Life Cycles (SDLC) merupakan metode pengembangan perangkat lunak yang dapat dijelaskan secara deskriptif maupun preskriptif tentang bagaimana perangkat lunak tersebut sedang atau akan dikembangkan (Ragunath, et al., 2010). Dalam SDLC, terdapat langkah-langkah untuk menganalisis, merancang, mengimplementasikan dan memelihara suatu perangkat lunak sistem informasi. Dalam SDLC sendiri terdapat beberapa teknik pengembangan sistem yang umum digunakan, sedangkan untuk penelitian ini akan menggunakan teknik Waterfall Model.

2.6 Waterfall Model

Waterfall Model merupakan salah satu model SDLC berbentuk linear sequensial yang diperkenalkan oleh Royce pada tahun 1970 (Mishra & Dubey, 2013). Model SDLC ini terdiri dari tahapan *requirements analysis*, *design*, *implementation*, *testing* dan *deployment & maintenance* yang berurutan sehingga ketika suatu fase telah diselesaikan maka fase tersebut tidak akan diulangi kembali dan proses pengembangan tidak akan berlanjut ke tahap berikutnya apabila fase sebelumnya belum sepenuhnya selesai.

Berdasarkan Sommerville (2007), *Waterfall Model* sebaiknya digunakan ketika kebutuhan dan persyaratan dari sistem telah dipahami dengan baik dan tidak mungkin berubah secara radikal selama pengembangan sistem. karena model SDLC ini menekankan perencanaan pada tahap awal, model ini memastikan dan menjelaskan apa saja kelebihan dan kekurangan dari rancangan perangkat lunak sebelum dikembangkan. Selain itu, adanya dokumentasi dan perencanaan yang intensif membuat model ini baik digunakan untuk proyek-proyek dimana *quality control* menjadi perhatian utama (Munassar & Govardhan, 2010).



Gambar 2.2 Diagram SDLC *Waterfall Model*

Sumber : Bassil, Y. (2012)

Berikut penjelasan dari Gambar 2.2 mengenai tahapan yang digunakan dalam *Waterfall Model* akan dijelaskan dalam beberapa sub-bab selanjutnya.

2.7 Tahap Analisis

Tahap Analisis atau *Analysis Phase* merupakan proses yang melibatkan proses analisis sehingga menghasilkan dokumen Software Requirements Specification (SRS) yang mana merupakan deskripsi lengkap dan menyeluruh dari perilaku suatu perangkat lunak yang akan dikembangkan. Fase ini melibatkan analisis kebutuhan bisnis dan kebutuhan sistem untuk menentukan persyaratan fungsional dan non-fungsional.

2.7.1 Analisis Proses Bisnis

Pada Tahap Analisis dibutuhkan analisis terhadap kebutuhan bisnis dari organisasi atau instansi yang terkait. Pada kegiatan analisis proses bisnis ini akan dilakukan pemetaan terlebih dahulu terhadap proses bisnis yang sedang berlaku dari organisasi terkait. Proses bisnis yang sedang berlaku saat ini di dalam organisasi terkait dapat disebut sebagai proses bisnis *as-is*. Dalam melakukan analisis proses bisnis *as-is*, pertama-tama dibutuhkan pengumpulan informasi mengenai pemangku kepentingan yang terlibat dalam suatu proses. Setelah informasi tersebut telah diketahui, selanjutnya dapat dilakukan pengumpulan informasi mengenai kondisi proses bisnis yang berjalan saat ini. Teknik wawancara dan observasi merupakan teknik yang paling tepat untuk memahami kondisi proses bisnis yang tengah berjalan saat ini (Brandenburg, 2017). Kemudian informasi mengenai proses bisnis yang ada dan pihak yang terlibat terhadap proses bisnis tersebut dapat digambarkan dan dimodelkan menggunakan *Business Process Model and Notation*. Setelah itu, informasi yang telah dikumpulkan dapat didokumentasikan ke dalam tabel analisis permasalahan sehingga dapat dilakukan analisis mengenai solusi dan perbaikan terhadap proses bisnis *as-is*. Pada Tabel 2.1 berikut merupakan contoh dari cara dalam melakukan analisis permasalahan.

Tabel 2.1 Analisis Permasalahan

Masalah	[Deskripsi masalah]
Mempengaruhi	[Pemangku kepentingan yang terpengaruh oleh masalah]
Dampak	[Dampak dari masalah]
Solusi	[Solusi beserta manfaatnya]

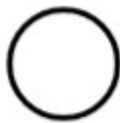
Sumber : Bittner (2002)


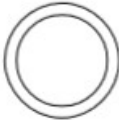


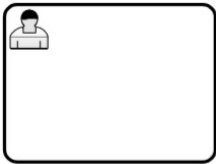

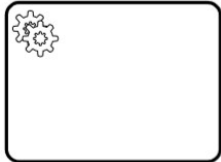

Kemudian setelah dilakukan analisis permasalahan dan dirumuskan solusinya, selanjutnya diperlukan pemodelan terhadap proses bisnis *to-be*. Proses bisnis *to-be* merupakan proses bisnis terbaru yang akan dijadikan sebagai perbaikan terhadap permasalahan yang ada dari organisasi terkait. Tujuan dari analisis proses bisnis *to-be* adalah untuk menjelaskan dan memberi gambaran kepada pemangku kepentingan organisasi terkait mengenai solusi dan perbaikan terhadap proses bisnis yang sedang berjalan, apakah telah memenuhi kebutuhan dari pemangku kepentingan yang terlibat dalam proses bisnis tersebut (Brandenburg, 2017).


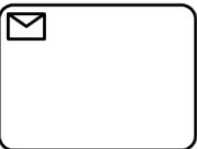
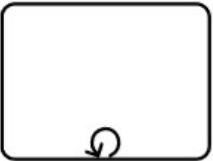
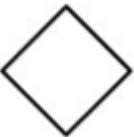



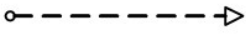

2.7.1.1 Business Process Model and Notations




Business Process Model & Notation (BPMN) merupakan sekumpulan diagram, yang dirancang untuk digunakan oleh orang-orang yang merancang dan mengelola proses bisnis (Object Management Group, 2011). BPMN menyediakan kemampuan kepada suatu organisasi untuk memahami prosedur bisnis internal dalam bentuk notasi grafis serta memberikan kemampuan kepada organisasi untuk menyampaikan prosedur tersebut dengan cara yang baku. BPMN sendiri dirancang untuk menyediakan standar mekanisme visualisasi proses bisnis yang didefinisikan dalam bahasa eksekusi proses bisnis yang optimal. Dalam pembuatan BPMN, diperlukan pemahaman mengenai beberapa bentuk penggambaran dari suatu proses bisnis. Bersamaan dengan pernyataan tersebut, tidak menutup kemungkinan juga sebenarnya dibutuhkan pemahaman yang beragam juga dari suatu proses yang sama seiring dengan berjalannya siklus *development, implementation, execution, monitoring* dan *analysis* dari suatu proses. Dalam merancang dan menggambarkan proses bisnis, dibutuhkan pengetahuan bagi perancang mengenai elemen-elemen yang digunakan pada BPMN. Berikut pada Tabel 2.2 merupakan beberapa penjelasan dari elemen-elemen yang digunakan dalam menggambarkan BPMN, antara lain:

Tabel 2.2 Notasi pada BPMN

No	Elemen	Deskripsi	Notasi
1	Event	Event adalah notasi yang digunakan untuk menggambarkan suatu peristiwa yang terjadi didalam suatu Process. Event mampu mempengaruhi alur dari suatu model dan umumnya memiliki penyebab dan akibatnya. Terdapat tiga jenis event berdasarkan kapan notasi ini digunakan pada suatu flow : Start, Intermediate, and End	

No	Elemen	Deskripsi	Notasi
2	Start Event	Start Event menunjukkan dimana Process tertentu atau Choreography akan dimulai	
3	Intermediate Event	Intermediate Event terjadi diantara Start Event dan End Event. Event jenis ini akan mempengaruhi alur dari suatu Process atau Choreography selain memulai atau mengakhiri Process	
4	End Event	End Event menunjukkan dimana suatu Process atau Choreography akan berakhir	
5	Activity	Activity adalah bentuk umum untuk Task yang dilakukan oleh organisasi dan digunakan pada Proses dan Choreography. Beberapa jenis dari Activity yang dapat digunakan pada model proses : Sub-Proses dan Task	
6	User Task	User Task menggambarkan Task yang dilakukan oleh seseorang, dengan bantuan dari aplikasi yang mendukung Process dan dikelola oleh komponen perangkat lunak	
7	Manual Task	Manual Task menggambarkan Task yang membutuhkan keterlibatan manusia untuk diselesaikan. Manual Task dilaksanakan tanpa menggunakan bantuan dari aplikasi eksekusi proses bisnis	
8	Service Task	Service Task menggambarkan Task yang sama sekali tidak membutuhkan interaksi dari manusia. Task dilakukan secara otomatis oleh semacam aplikasi atau layanan eksternal	
9	Script Task	Script Task merepresentasi kode yang dijalankan oleh <i>process business engine</i> pada <i>process engine</i> . Pelaksana <i>process</i> mengartikan script dari Task dalam bahasa yang dapat diterjemahkan oleh <i>engine</i> .	
10	Send Task	Send Task merupakan Task yang dirancang untuk mengirimkan Message kepada Participant eksternal atau Pool lain	

No	Elemen	Deskripsi	Notasi
			
11	Receive Task	Receive Task merupakan Task yang dirancang untuk menunggu dan menerima Message yang datang dari Participant eksternal	
12	Loop Characteristics Task	Loop Characteristics memberikan sebuah Activity perilaku untuk melakukan perulangan. Perilaku perulangan didasarkan kepada kondisi Boolean, yang kondisi tersebut selalu diperiksa selama memenuhi kondisi <i>true</i> maka Activity akan tetap diulang	
13	Gateway	Gateway digunakan untuk mengendalikan perpecahan dan pertemuan dari Sequence Flow dalam suatu Proses dan pada Choreography. Gateway dapat menentukan percabangan dan penggabungan dari jalur yang dilalui. Marker didalam Gateway menentukan jenis kendali perilaku	
14	Exclusive Gateway	Exclusive Gateway digunakan untuk membuat jalur alternative didalam alur Process. Alur Process yang menggunakan Exclusive Gateway hanya membolehkan satu jalur yang bisa diambil	
15	Inclusive Gateway	Inclusive Gateway digunakan untuk membuat jalur alternative sekaligus jalur parallel didalam alur Process. Dalam Inclusive Gateway seluruh kondisi akan dievaluasi, dan memungkinkan lebih dari 1 jalur untuk dilalui	
16	Sequence Flow	Sequence flow digunakan untuk menunjukkan urutan dari Activity yang akan dikerjakan dalam suatu Proses dan Choreography	
17	Message Flow	Message Flow digunakan untuk menunjukkan alur antar dua Participant yang disediakan untuk mengirim dan menerima pesan	
18	Pool	Pool adalah sebuah penggambaran grafis dari sebuah Participant di dalam Collaboration. Pool juga berperan sebagai wadah untuk membagi sekumpulan Activity dari Pool yang lain.	

No	Elemen	Deskripsi	Notasi
19	Lane	Lane adalah sub-bagian didalam sebuah Process dan terkadang didalam sebuah Pool. Lane digunakan untuk mengelola dan mengkatagorikan Activity	
20	Data Object	Data Object menyediakan informasi mengenai Activity apa yang butuh dikerjakan atau apa yang Activity hasilkan	
21	Message	Message digunakan untuk menggambarkan konten dari komunikasi antar dua Participant	

Sumber : Object Management Group (2011)

2.7.2 Analisis Kebutuhan Fungsional dan Non-Fungsional

Kebutuhan fungsional merupakan pernyataan dari fungsi-fungsi yang harus dimiliki dan dapat dilakukan oleh sebuah sistem, bagaimana sistem tersebut harus dapat memberi tanggapan berdasarkan masukan yang telah diberikan dan bagaimana perilaku dari sistem dalam situasi tertentu. Terkadang kebutuhan fungsional secara eksplisit juga menjelaskan tentang apa saja yang seharusnya tidak dilakukan oleh sistem (Sommerville, 2011). Sedangkan kebutuhan non-fungsional merupakan pernyataan dari batasan-batasan atau fungsi-fungsi yang dapat ditawarkan dan dilakukan oleh sistem. Batasan-batasan tersebut dapat berupa batasan waktu, batasan mengenai proses pengembangan, batasan mengenai keamanan, serta batasan yang diberlakukan sebagai ketentuan. Kebutuhan non-fungsional sering kali tidak hanya berlaku untuk masing-masing fitur dari sistem, melainkan berlaku untuk sistem secara keseluruhan (Sommerville, 2011).

Langkah pertama yang harus dilakukan dalam melakukan analisis kebutuhan fungsional dan non-fungsional adalah dengan mengumpulkan dan mencatat kebutuhan. Pendekatan yang dapat dilakukan untuk melakukan analisis kebutuhan adalah dengan menggunakan dokumen yang terdiri atas daftar kebutuhan-kebutuhan fungsional dan batasan-batasan yang menjadi kebutuhan non-fungsional (Sommerville, 2011). Sedangkan menurut Wazlawick (2014), pendekatan lain yang dapat dilakukan untuk melakukan analisis kebutuhan fungsional dan non-fungsional adalah dengan menggunakan *Use Case Diagram*.

2.8 Tahap Perancangan

Tahap Perancangan atau *Design Phase* merupakan proses yang melibatkan pengembang perangkat lunak dan perancang untuk menentukan rencana dari solusi yang mencakup perancangan arsitektur perangkat lunak, skema konseptual basis data, definisi struktur data serta desain *graphical user interface*. Dalam melakukan dan membuat perancangan arsitektur perangkat lunak, skema konseptual basis data dan definisi struktur data, pengembang dapat menggunakan

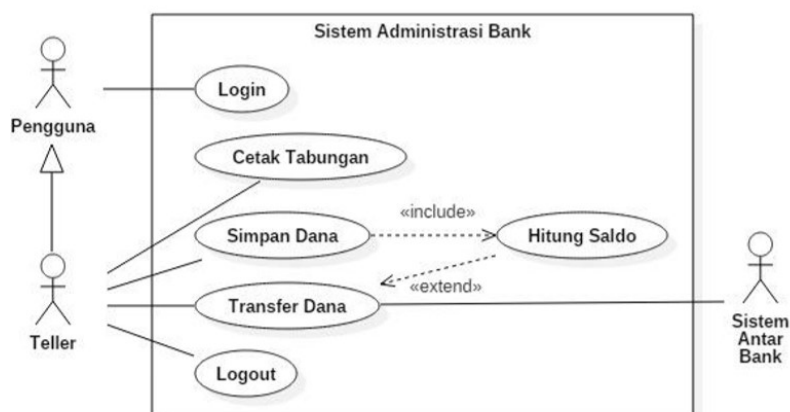
bahasa pemodelan yang terintegrasi untuk keperluan pengembangan perangkat lunak. Kumpulan bahasa pemodelan ini sering disebut dengan *Unified Modelling Language*.

2.8.1 Unified Modelling Language

Unified Modelling Language (UML) adalah kerangka kerja yang menjadi standar untuk digunakan dalam pembuatan *blueprint* dari perangkat lunak. UML dapat digunakan untuk memvisualisasikan, menjabarkan spesifikasi, melakukan konstruksi, dan membuat dokumentasi artifak dari sistem perangkat lunak (Pressman, 2010). Berdasarkan penjelasan tersebut maka kegunaan dari UML adalah untuk memberi gambaran secara jelas tentang bagaimana kerangka dan isi dari perangkat lunak lalu membuat dokumentasi ke dalam bentuk diagram dari seluruh proses perancangan perangkat lunak. UML memiliki beberapa diagram yang dapat digunakan untuk menggambarkan sistem yang sedang dirancang, dan beberapa diagram UML yang akan digunakan untuk perancangan sistem dalam penelitian ini antara lain terdiri dari *Use Case Diagram*, *Use Case Scenario*, *Sequential Diagram* dan *Class Diagram*.

2.8.1.1 Use Case Diagram

Use Case Diagram merupakan sebuah diagram yang digunakan untuk mendeskripsikan perilaku dari sebuah sistem dibawah kondisi tertentu dalam menanggapi interaksi permintaan dari salah satu *stakeholdernya* (Pressman, 2010). Pengertian lain dari *use case diagram* adalah sebuah penggambaran interaksi yang terjadi antara pengguna dengan sistem (Kurniawan, 2018). Berdasarkan pengertian yang telah dijelaskan, *use case diagram* merupakan diagram dari sudut pandang pengguna yang digunakan untuk mendeskripsikan fungsi-fungsi perilaku dan menunjukkan bagaimana interaksi antara *stakeholder* baik berupa pengguna maupun berupa perangkat lain dengan suatu sistem. Berikut pada Gambar 2.3 merupakan sebuah contoh penggambaran *use case diagram* sederhana dari Sistem Administrasi Bank:





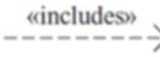


Gambar 2.3 Contoh use case diagram

Sumber : Kurniawan (2018)

Dalam menggambarkan sebuah use case, langkah pertama yang perlu dilakukan adalah mendefinisikan aktor-aktor yang akan terlibat didalam interaksi sebuah sistem. Setelah aktor didefinisikan, maka langkah selanjutnya adalah mendefinisikan task-task dan fungsi-fungsi utama yang akan dilakukan oleh aktor tersebut, interaksi-interaksi apa saja yang memungkinkan untuk terjadi. Selain itu juga perlu didefinisikan mengenai informasi sistem apa saja yang akan dibutuhkan, dihasilkan atau diubah oleh aktor tersebut. Didalam *use case diagram* terdapat 5 elemen penyusun yang dapat digunakan. Penjelasan mengenai elemen-elemen penyusun tersebut akan dijelaskan pada Tabel 2.3:

Tabel 2.3 Simbol Use Case Diagram

No	Tipe	Deskripsi	Simbol
1	<i>Use Case</i>	<i>Use Case</i> mewakili fungsi-fungsi dari sistem yang dilakukan dan dirasakan oleh pengguna dan setiap UC bisa dijalankan oleh lebih dari satu aktor.	
2	Aktor	Aktor adalah segala sesuatu yang berinteraksi langsung dengan sistem, bisa merupakan orang beserta perannya atau sistem komputer yang lain.	
3	<i>Association</i>	<i>Association</i> merupakan sebuah penghubung antara aktor dengan use case. <i>Association</i> menandakan bahwa terdapat komunikasi antar aktor dengan sebuah <i>use case</i> , seperti mengirim atau menerima pesan.	
4	<i>Extend</i>	<i>Extend</i> menyatakan bahwa fungsionalitas dari base <i>use case</i> bisa diperluas oleh <i>supplier use case</i> , jika dibutuhkan, di dalam eksekusi alternatif yang ada pada <i>use case scenario</i> dari base <i>use case</i> .	
5	<i>Include</i>	<i>Include</i> menyatakan bahwa fungsionalitas dari base <i>use case</i> selalu hanya bisa dipenuhi dengan bantuan dari <i>supplier use case</i> di dalam eksekusi alur utama yang ada pada <i>use case scenario</i> dari base <i>use case</i> .	

Sumber : Kurniawan (2018)

2.8.1.2 Use Case Scenario

Use Case Scenario merupakan penjelasan secara tekstual dari sekumpulan skenario urutan aksi yang dilakukan aktor ketika melakukan interaksi dengan sistem, baik yang berhasil maupun yang gagal (Kurniawan, 2018). Penjelasan secara tekstual dalam pembuatan use case scenario terdiri dari beberapa format yang dapat disesuaikan dengan kebutuhannya, yaitu singkat, informal, atau lengkap dan dapat dijelaskan ke dalam bentuk tabel dengan 1 kolom atau 2 kolom.

Pada format singkat, penjelasan terdiri dari 1 paragraf yang mengacu hanya pada skenario berhasil. Pada format informal, penjelasan terdiri dari beberapa paragraf yang mencakup skenario yang berhasil maupun yang gagal. Sedangkan pada format lengkap, penjelasan terdiri dari rincian lengkap disertai dengan bagian-bagian pendukung yang penting.


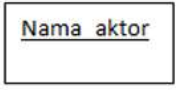

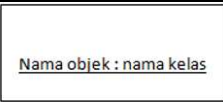


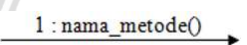
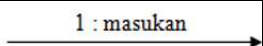
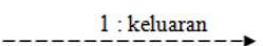
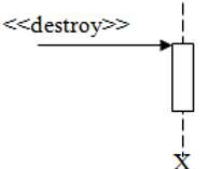
Pada penelitian ini format use case scenario yang akan digunakan adalah menggunakan format lengkap. Bagian-bagian penting dari penulisan dengan format lengkap antara lain harus terdiri dari aktor primer, kondisi awal, alur utama, alur alternatif dan kondisi akhir (Kurniawan, 2018). Aktor primer merupakan aktor yang memulai layanan dari sistem untuk mencapai tujuan dari aktor tersebut. Kondisi awal merupakan kondisi yang harus terpenuhi sebelum sebuah *use case* bisa dimulai oleh aktor primer dan bisa terdiri dari beberapa keadaan. Alur utama merupakan jalur interaksi yang terdiri dari 1 jalur untuk mengarahkan pada skenario yang berhasil sehingga tujuan aktor dapat terpenuhi. Alur alternatif merupakan beberapa jalur pencabangan atau jalur skenario gagal dari interaksi yang terjadi antar aktor dengan sistem sehingga menyebabkan tujuan aktor menjadi tidak terpenuhi. Dalam menjelaskan alur utama dan alur alternatif diperlukan penjelasan secara lengkap mengenai gambaran interaksi antara aktor dengan sistem yang dilihat dari dua sisi, yaitu dari sisi aktor dan dari sisi sistem. Selain itu, dalam aliran utama dan aliran alternatif harus mampu memberikan penjelasan dari tanggung jawab yang harus dilakukan oleh sistem, bukan bagaimana sistem melakukannya. Sedangkan kondisi akhir merupakan kondisi yang harus terjadi ketika sebuah *use case* berhasil dijalankan secara lengkap dan menunjukkan tujuan yang ingin dicapai oleh aktor primer.

2.8.1.3 Sequence Diagram

Sequence Diagram merupakan diagram perancangan sistem pada *Unified Modelling Language* (UML) yang berfungsi untuk memodelkan interaksi antara aktor dengan objek-objek hingga antar objek-objek itu sendiri dalam sebuah sistem (Sommerville, 2011). Perancangan menggunakan *sequence diagram* dibutuhkan untuk menyediakan penjelasan dan pemahaman yang lebih rinci mengenai interaksi-interaksi antara aktor dengan sistem berdasarkan dari diagram *use case* yang telah dibuat pada tahap analisis. Oleh karena itu dalam menggambarkan *sequence diagram* harus diawali dengan menentukan objek-objek apa saja yang terlibat di dalam *use case* beserta *method-method* yang dimiliki. Membuat *sequence diagram* juga membutuhkan skenario yang terdapat pada *use case*. Jumlah *sequence diagram* yang harus digambar adalah sedikitnya sebanyak pendefinisian *use case* yang memiliki proses sendiri atau setidaknya semua *use case* yang telah didefinisikan interaksi dari jalannya pesan yang sudah dicakup pada *sequence diagram*. Jika semakin banyak *use case* yang telah didefinisikan maka akan menyebabkan *sequence diagram* yang harus dibuat juga akan bertambah banyak.

Pada Tabel 2.4 berikut merupakan penjelasan mengenai simbol-simbol yang digunakan dalam *sequence diagram*:

Tabel 2.4 Simbol *Sequence Diagram*






No	Tipe	Deskripsi	Simbol
1	Aktor	Orang, proses atau system lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor	 Nama_aktor Atau 
2	Garis Hidup / Lifeline	Menyatakan kehidupan suatu objek	
3	Objek	Menyatakan objek yang berinteraksi pesan	
4	Waktu Aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya	
5	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat	
6	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri	
7	Pesan tipe <i>send</i>	Menyatakan suatu objek mengirimkan data/masukan/ informasi ke objek lainnya	
8	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang sudah menjalankan suatu operasi atau metode menghasilkan kembalian ke objek tertentu	
9	Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri	

Sumber: Handika (2016)

2.8.1.4 Class Diagram

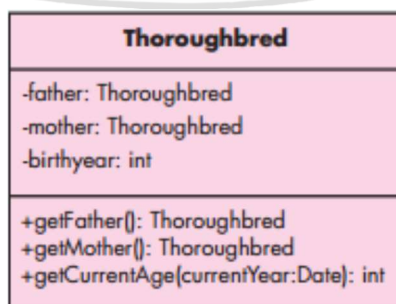
Class diagram digunakan untuk menunjukkan *class* atau *object* apa saja yang saling berkaitan, berhubungan serta bagaimana *class* dan *object* tersebut saling bertukar informasi di dalam suatu sistem (Booch, et al., 2005). Elemen yang membawa peran paling penting dalam *class diagram* adalah *class* yang berisi atribut dan fungsi yang dimiliki oleh *class* tersebut serta bagaimana hubungan antar satu *class* dengan *class* yang lainnya. Berikut pada tabel 2.5 merupakan penjelasan mengenai simbol yang digunakan pada *class diagram*, antara lain:

Tabel 2.5 Penjelasan simbol Class Diagram

No	Tipe	Deskripsi	Simbol
1	<i>Class</i>	<i>Class</i> merupakan sebuah <i>template</i> dimana objek dibuat. <i>Class</i> mendefinisikan atribut, operasi dan <i>instance</i> .	
2	<i>Association</i>	Relasi yang menunjukkan hubungan antara dua <i>class</i> .	
3	<i>Generalization</i>	Relasi yang menunjukkan pewarisan dari <i>class</i> utama (<i>parent</i>) ke <i>class</i> anak (<i>child</i>)	
4	<i>Dependency</i>	Relasi yang menunjukkan suatu <i>class</i> bergantung pada <i>class</i> yang lain	
5	Agregasi / <i>Aggregation</i>	Relasi yang menunjukkan sebuah elemen yang terdiri dari beberapa komponen kecil	

Sumber : Sukanto & Shalahudin (2014)

Berikut pada Gambar 2.4 merupakan contoh dari penggambaran *class diagram* beserta dengan atribut, fungsi serta jenis fungsi-fungsi yang dapat dilakukan oleh kelas Thoroughbred:



Gambar 2.4 Class diagram dari kelas Thoroughbred

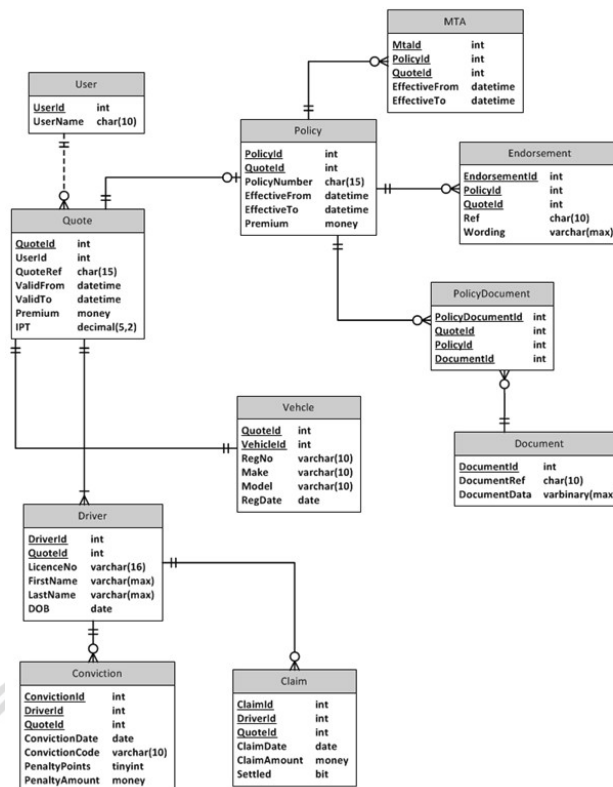
Sumber : Pressman (2010)

Kotak teratas dari Gambar 2.4 berisi nama dari *class* yang pada umumnya mewakili sebuah obyek yang ada pada sistem. Kotak dibawahnya berisikan penjelasan nama dan tipe data dari atribut-atribut yang dimiliki oleh obyek kelas tersebut. Sedangkan kotak terakhir yang paling bawah berisikan segala macam fungsi yang dimiliki dan dapat dilakukan oleh obyek dari instansiasi kelas tersebut serta menjelaskan tipe data keluaran dari fungsi tersebut. Atribut dan fungsi memiliki beberapa tipe hak akses enkapsulasi yang dinotasikan sebagai +, ~, #, - secara urut mewakili public, package, protected, dan private (Pressman, 2010).

2.8.2 Physical Data Model

Physical Data Model (PDM) merupakan sebuah model spesifik untuk basis data yang merepresentasikan objek data relasional seperti tabel, kolom, *primary key* dan *foreign key* beserta dengan relasinya. Sebuah PDM dapat digunakan untuk menghasilkan pernyataan *Data Definition Language* (DDL) yang mana kemudian dapat diterapkan ke dalam *server* basis data (IBM, 2018).

Berikut pada Gambar 2.5 merupakan contoh dari penggambaran PDM dan penjelasannya adalah sebagai berikut. *MTA, Policy, Endorsement, PolicyDocument, Document, Quote, User, Vehicle, Driver, Claim* dan *Conviction* merupakan contoh penggambaran dan penamaan dari objek tabel. Penulisan didalam masing-masing objek tabel merupakan penjelasan dari atribut atau kolom yang dimiliki dari masing-masing tabel tersebut, diikuti dengan tipe data dari masing-masing kolom. Penulisan atribut dengan garis bawah seperti UserId menunjukkan bahwa kolom tersebut merupakan *primary key* dari tabel *User*. Sedangkan penulisan atribut dengan garis bawah seperti PolicyId di tabel *Endorsement* menunjukkan bahwa kolom tersebut merupakan *foreign key* dari tabel *Endorsement* yang mereferensi kepada kolom PolicyId pada tabel *Policy*. Selain itu, penggambaran relasi antar tabel juga ditunjukkan dengan menggunakan garis utuh dan garis putus-putus. Garis putus-putus antara tabel *User* dengan tabel *Quote* menjelaskan bahwa relasi antar dua tabel tersebut merupakan relasi lemah (*non-identifying*), yang berarti bahwa keberadaan data pada tabel *Quote* tidak bergantung kepada keberadaan dari data pada tabel *User*. Sedangkan garis utuh antara tabel *Quote* dengan tabel *Policy* menjelaskan bahwa relasi antar dua tabel tersebut merupakan relasi kuat (*identifying*), yang berarti bahwa keberadaan data pada tabel *Policy* bergantung kepada keberadaan dari data pada tabel *Quote*.



Gambar 2.5 Contoh Physical Data Model

Sumber : Rayan (2015)

Berdasarkan penjelasan sebelumnya mengenai Tahap Perancangan pada metode SDLC *Waterfall Model*, bahwa tahap ini juga melibatkan proses perancangan skema basis data dan definisi struktur data. Dalam menerapkan proses perancangan skema basis data dan definisi struktur data pada basis data relasional, pendekatan perancangan berorientasi objek dapat digunakan. *Class* dan hubungan antar *class*nya dapat dipetakan ke dalam table-table di rancangan basis data relasional. Dalam penerapan perancangan menggunakan pendekatan ini, *persistent class* dan atributnya dipetakan menjadi entitas. Hubungan asosiasi-asosiasi dipetakan menjadi relasi non-spesifik di dalam entity relationship diagram (ERD). Jenis atribut kelas dipetakan ke tipe elemen data yang sesuai pada ERD. Diagram ER memungkinkan perancang basis data untuk melakukan pemeriksaan normalisasi terhadap skema basis data serta mengimplementasikan *foreign key*. Dari diagram ER yang telah dibuat, perancang dapat membuat *physical data model* untuk berbagai sistem manajemen basis data relasional yang diinginkan (IBM, 2018).

2.9 Tahap Implementasi

Tahap Implementasi atau *Implementation Phase* merupakan proses yang mengacu pada realisasi persyaratan bisnis dan *design specification* menjadi komponen program, basis data, *website*, atau perangkat lunak konkrit yang bisa digunakan melalui implementasi pemrograman. Dengan kata lain, tahap ini

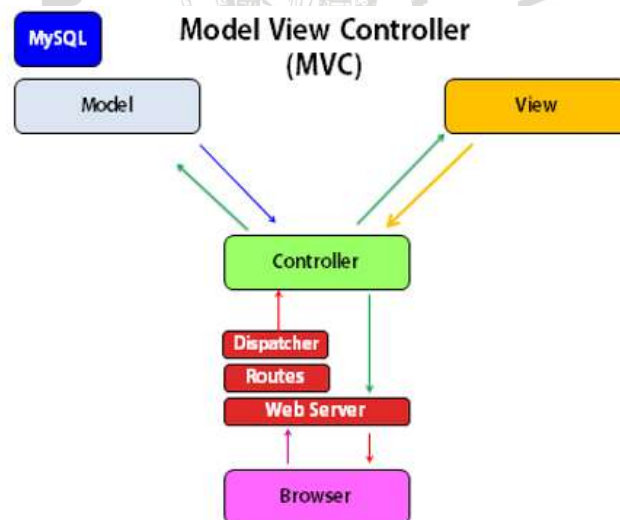
adalah proses untuk mengubah keseluruhan persyaratan dan *blueprint* sistem menjadi lingkungan produksi.

2.9.1 Framework CodeIgniter

Framework merupakan sekumpulan *library* yang terorganisir dalam suatu desain arsitektur yang mampu untuk memberikan kecepatan, ketepatan dan kenyamanan dalam pengembangan suatu aplikasi (Hustinawati, et al., 2014), serta terdiri dari elemen *Architecture*, *File Library* dan *Methodology*.

CodeIgniter adalah sebuah *framework* aplikasi web yang dikenalkan pada tanggal 28 Februari 2006 digunakan untuk membangun suatu aplikasi PHP dinamis dan bersifat *open source*. Alasan yang mendasari CodeIgniter menjadi suatu *framework* yang bersifat *open source* adalah agar dapat membantu pengembangan dari pengembang CodeIgniter dalam mengerjakan suatu aplikasi menjadi lebih cepat dibandingkan dengan menulis seluruh kode program dari awal. Selain itu, CodeIgniter sendiri digunakan karena *framework* ini memiliki beberapa karakteristik sebagai berikut :

1. Bersifat fleksibel dan ringan sehingga memudahkan untuk dipelajari, dimodifikasi, dan diintegrasikan dengan *library* dan *helper*
2. Menggunakan pola *Model-View-Controller* (MVC) yang memungkinkan pemisahan kode program bagian logika dengan tampilan, sehingga mampu menghasilkan struktur kode program yang lebih terstruktur dan lebih baku



Gambar 2.6 Struktur Kerja CodeIgniter

Sumber : Hustinawati, et al. (2014)

Struktur kerja dari *CodeIgniter* seperti yang digambarkan pada Gambar 2.6, diawali dari *browser* yang akan berinteraksi dengan *controller* melalui *web server*, *routes* lalu kemudian dengan *dispatcher*. *Controller* disini berperan untuk menerima dan membalas seluruh permintaan dari *browser*. Untuk keperluan mendapatkan dan melakukan perubahan data dari basis data dan sumber data lain, *controller* akan melakukan *request* ke *model*. Sedangkan untuk keperluan

tampilan antarmuka pengguna *controller* akan melakukan request ke *view*. Ketika *browser* melakukan *request* sebuah halaman *web*, *router* akan mencari *controller* yang bertanggung jawab untuk menangani permintaan tersebut, kemudian *controller* tersebut akan menggunakan *model* untuk mengakses data yang diperlukan lalu menggunakan *view* untuk menampilkan data yang telah diakses.

Pada penelitian ini, *Framework* CodeIgniter akan digunakan sebagai kerangka kerja untuk membangun *website operator* yang akan digunakan oleh pegawai Perlengkapan. Pengembangan *website operator* dengan *Framework* CodeIgniter ini akan menggunakan PHP sebagai bahasa pemrograman *web* sisi *server*nya. Sedangkan bahasa pemrograman *web* yang akan digunakan untuk sisi *client*nya antara lain adalah *Hypertext Markup Language* (HTML) dan *Cascading Style Sheet* (CSS). HTML digunakan untuk menyusun struktur dari halaman *web* yang akan dibangun dan CSS digunakan untuk menyusun tata letak dari halaman *web* tersebut. *Website operator* ini fungsinya adalah untuk menerima laporan keluhan mengenai sarana prasarana yang disampaikan oleh pihak pelapor dari civitas akademik FILKOM serta mengirimkan notifikasi perubahan status penanganan laporan keluhan tersebut kepada aplikasi perangkat bergerak yang digunakan oleh pihak pelapor. Dalam menampilkan dan melakukan perubahan data mengenai laporan keluhan yang masuk atau data mengenai pengguna pelapor dan *operator*, *Website operator* perlu melakukan koneksi terlebih dahulu kepada suatu basis data sehingga data yang tersimpan dapat terjaga keutuhannya dan dapat diakses secara real-time.

2.9.2 Aplikasi Android

Android merupakan sebuah sistem operasi dan *platform* perangkat lunak untuk perangkat bergerak berbasis kernel *Linux* yang awalnya dikembangkan oleh Google dan kemudian dikembangkan oleh *Open Handset Alliance* (Bhardwaj, et al., 2013). *Android* merupakan perangkat lunak *open-source* yang terdiri dari sistem operasi, *middleware*, serta *key application* berbasis *Linux* dan Java yang dapat diunduh secara gratis. *Android* memungkinkan *developer* untuk menuliskan kode program dan membuat aplikasi perangkat bergerak *native* dalam bahasa Java serta mengendalikan fitur-fitur dari perangkat tersebut melalui *library* Java yang dikembangkan oleh Google. Pertimbangan dari pengembangan sistem menggunakan *platform* aplikasi *native* *Android* dibandingkan dengan aplikasi *web* adalah karena *platform* ini mampu memberikan kinerja dan fungsionalitas yang lebih baik terkait dengan akses langsung kepada *hardware* dari perangkat (Jobe, 2013).

Pada penelitian ini, aplikasi perangkat bergerak berbasis *Android* yang akan dibuat digunakan oleh pihak pelapor dari civitas akademik FILKOM untuk melaporkan masalah mengenai sarana prasarana. Laporan masalah mengenai sarana prasarana dari aplikasi perangkat bergerak *Android* ini akan dikirimkan ke *website operator* yang digunakan oleh pegawai Perlengkapan. Sehingga nantinya laporan yang masuk tersebut dapat ditindak lanjuti dan diubah status penanganannya oleh pegawai Perlengkapan agar *website operator* dapat

mengirimkan informasi berupa notifikasi mengenai perubahan status penanganan dari laporan yang telah dilaporkan. Dalam mengirimkan laporan dari aplikasi perangkat bergerak ke *website operator* dan notifikasi dari *website operator* ke aplikasi perangkat bergerak, akan dibutuhkan komponen perangkat lunak lain berupa sistem basis data sebagai sistem penyimpanan data mengenai laporan dan data mengenai pengguna. Selain itu juga dibutuhkan *web service* sebagai perantara antara aplikasi *Android* agar dapat menyimpan data ke dalam sistem basis data serta mengakses data dari sistem basis data.

2.9.3 Web Service

Web service merupakan komponen perangkat lunak yang disimpan dalam suatu perangkat komputer dan dapat diakses oleh suatu aplikasi, komponen *software* lain atau perangkat komputer lain melalui suatu jaringan (Deitel & Deitel, 2012). *Web service* memungkinkan sistem yang berbeda-beda untuk saling berkomunikasi dengan mengirimkan sebuah pesan satu dengan lainnya (Abeyasinghe, 2008). Perangkat yang menyediakan *service* dapat disebut dengan *web service host*. Cara kerja dari *web service* adalah aplikasi *client* berkomunikasi dengan mengirimkan *request* kepada *web service host* melalui suatu jaringan lalu memprosesnya kemudian mengembalikannya berupa *response* kepada aplikasi *client*. Dalam penelitian ini, *web service* digunakan untuk menghubungkan aplikasi perangkat bergerak pelapor sebagai aplikasi *client* dengan halaman situs *web* pegawai perlengkapan sebagai *web service host* melalui jaringan *internet*.

Dalam mengimplementasikan *web service*, terdapat berbagai macam teknologi yang mampu mendukungnya. Teknologi yang paling sering digunakan untuk keperluan ini antara lain adalah menggunakan *Representational State Transfer*(REST) serta *Simple Object Access Protocol*(SOAP) sebagai alternatif yang juga sering digunakan. REST *Web Service* lebih sering digunakan karena penggunaannya yang lebih sederhana, lebih mudah, lebih efisien biaya pengiriman serta menggunakan teknologi berbasis web yang ekstensif seperti HTTP yang telah diketahui dan lebih populer di kalangan pengembang aplikasi *web*. Cara kerja dari REST *Web Service* adalah dengan menerapkan prinsip-prinsip dari *World Wide Web*, dalam artian bahwa sebuah sistem di tempat yang berbeda menawarkan akses kepada aplikasi *client* seperti *browser* atau *mobile apps* ke halaman penyedia *resource* dengan mengandalkan serangkaian tindakan standar seperti GET, POST, PUT, DELETE (Arroqui, et al., 2012). Kemudian dalam mengirimkan pesan atau *resource*, REST *web service* menggunakan teknologi yang sudah menjadi standar seperti HTTP, HTML, XML dan JSON.

2.9.4 JSON

JavaScript Object Notation(JSON) merupakan *format* representasi dan pertukaran data berbasis teks yang digunakan untuk mewakili suatu objek sebagai kumpulan nilai dalam bentuk *String* (Deitel & Deitel, 2012). JSON merupakan *format* representasi sederhana yang membuat suatu objek dapat mudah dibaca, dibuat atau diurai. JSON menggunakan *format* teks biasa untuk mempermudah pengiriman, pengolahan dan pertukaran informasi antar layanan dan aplikasi yang

berbeda-beda (Santos & Serrao, 2016). Penggunaan dari JSON melalui HTTP telah menjadi suatu solusi teknologi yang memungkinkan untuk diberlakukannya pertukaran informasi antara layanan berbasis *web* dengan suatu aplikasi. JSON banyak digunakan untuk mendukung komunikasi antar beberapa *Application Programming Interface* (API) yang tersedia di internet saat ini, terutama keperluannya untuk mendukung fungsi dari REST *web service*.

JSON membuat *format* representasi suatu objek yang bisa mengandung kumpulan dari beberapa nilai-nilai yang dapat berupa data *string*, angka dan *boolean*. Kumpulan nama dan nilai yang dikandung dari setiap objek JSON diletakkan dan dituliskan di dalam tanda kurung kurawal seperti pada Gambar 2.7:

```
{
  "id" : "1",
  "f_name" : "Adam",
  "l_name" : "Smith",
  "skills" : [
    "web development",
    "ux design",
    "data science"
  ],
  "team": "engineering"
}
```

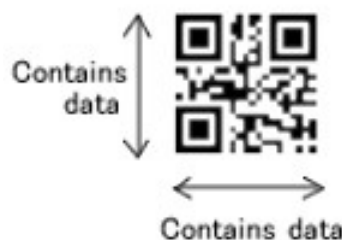
Gambar 2.7 Contoh representasi objek JSON

Sumber : www.json.com (2013)

Pada Gambar 2.8 dijelaskan bahwa selain nilai yang dikandung dari setiap objek JSON diletakkan di dalam tanda kurung kurawal, masing masing atribut dari nilai yang dikandung oleh suatu objek diwakilkan oleh *keys* yang dituliskan dalam bentuk data *string*. Contoh *keys* yang dimaksud antara lain adalah "id", "f_name", "l_name", "skills" dan "team". Sedangkan untuk penulisan nilai *array* pada JSON di letakkan didalam tanda kurung siku atau seperti yang dicontohkan nilai yang dikandung pada key "skills". Selain itu, atribut key "skills" pada Gambar 2.8 juga menjelaskan bahwa struktur nilai seperti *string*, angka, *boolean*, *array* atau objek lain yang dikandung dari setiap objek JSON juga dapat dibuat bersarang. Pada atribut key "skills" mengandung sebuah *array* JSON yang mengandung 3 buah *string* lain.

2.9.5 Quick Response Code (QR Code)

Quick Response Code (QR Code) adalah sebuah kode dengan wujud matriks 2 dimensi yang termasuk dalam sekumpulan kode yang dapat dibaca oleh mesin tertentu, yang semua kode-kode tersebut sering disebut sebagai *barcode*, terlepas dari bagaimana bentuknya baik terdiri dari sekumpulan garis vertikal, kotak atau bentuk lainnya (Denso ADC, 2011). QR Code dapat menyimpan suatu informasi dengan cara memperhatikan elemen elemen gelap dan terang yang disebut sebagai "*modules*" yang terdapat dalam susunan secara horizontal serta vertical. QR Code dirancang berbentuk persegi yang berisi kotak-kotak berwarna didalamnya dan memiliki tiga buah persegi di ketiga sudutnya. Pada Gambar 2.8 merupakan contoh penggambaran visual dari QR Code:



Gambar 2.8 Contoh QR Code

Sumber : Denso ADC (2011)

Pada awalnya, *QR Code* dikembangkan untuk melakukan pelacakan komponen otomotif melalui sistem dalam proses manufaktur dan rantai suplai distribusi. Kemudian penggunaan *QR Code* sendiri akhirnya mulai berkembang dari sistem lama yang masih menggunakan *barcode* tradisional, sampai ke beberapa sistem yang baru dikembangkan. Saat ini, *QR Code* sendiri sudah banyak digunakan pada *industry Marketing, Manufacturing, Warehousing and Logistic* serta pada *Office Automation*. Beberapa contoh pengaplikasian *QR Code* yang memiliki tujuan yang serupa terdapat pada industri *Warehousing and Logistic* dan *Manufacturing*, yaitu berupa pelacakan barang atau produk serta informasi detail dari barang tersebut, terlepas dari selengkap apa detail informasi yang terkandung dalam *QR Code* tersebut.

QR Code menjadi suatu teknologi yang semakin sering digunakan karena proses pembuatan yang sederhana, minimnya kendala untuk menggunakannya serta rendahnya biaya yang dibutuhkan untuk mengembangkan serta mengadopsi suatu sistem yang memanfaatkan teknologi ini (Ashford, 2010). Alasan lain yang menjadi keuntungan dari penggunaan *QR Code* adalah proses pemindaian yang cepat dan kapasitas penyimpanan data yang besar. Selain penyimpanan data yang besar, *QR Code* memiliki ukuran hingga 10% lebih kecil dibandingkan dengan *barcode* yang memiliki kandungan data yang sama. Lalu *QR Code* dapat dibaca dari segala arah dalam lingkup sudut 360 derajat. Serta *QR Code* memiliki kemampuan untuk mengoreksi kesalahan dengan tingkat kesalahan hingga 30%.

QR Code menyediakan metode yang mudah, murah dan aman untuk mengirimkan informasi dalam *format "push"* kepada pihak yang mampu membaca symbol kode tersebut (Lorenzi, et al., 2014). *Library open source* telah tersedia untuk menghasilkan *QR Code* dari berbagai macam sumber data selama data tersebut dapat masuk ke sejumlah karakter *alphanumeric* yang tetap, tergantung dari versi *QR Code* yang digunakan. Jika sebuah aplikasi dapat membaca versi tertentu dari data *QR Code*, aplikasi individual lain dapat memutuskan bagaimana untuk menangani data yang telah diekstraksi dari kode tersebut. Hal ini memungkinkan aplikasi *smartphone* untuk menguraikan data yang masuk dari *QR Code* lalu menggunakannya dengan cara yang diinginkan oleh pembuat program. Selain itu, dengan menggunakan *QR Code* maka pengguna tidak diharuskan untuk melakukan masukan data secara manual sehingga akurasi data masukan bisa semakin meningkat.

Pada penelitian ini, *QR Code* digunakan untuk membantu proses identifikasi aset sarana prasarana di aplikasi *android* pelapor. Dalam implementasinya *QR Code* perlu di *generate* terlebih dahulu di sisi *operator* pegawai sarana prasarana, kemudian setelah itu dicetak dalam bentuk kartu yang ditempelkan pada tiap sarana prasarana. Apabila ada pelapor yang ingin melakukan identifikasi suatu sarana prasarana, maka pada aplikasi *android* pelapor dapat memindai *QR Code* yang terdapat pada sarana prasarana tersebut sehingga informasi mengenai barang tersebut bisa didapatkan pelapor. Agar fitur untuk melakukan pemindaian pada aplikasi *android* bisa dilakukan, aplikasi tersebut perlu mengimplementasikan *ZXing library*. *ZXing* atau *Zebra Crossing* merupakan *library* dari bahasa pemrograman *Android Java* yang digunakan untuk melakukan pemindaian *barcode*.

2.10 Tahap Pengujian

Tahap Pengujian atau *Testing Phase* merupakan proses untuk memeriksa apakah solusi yang ditawarkan perangkat lunak tersebut telah memenuhi persyaratan dan spesifikasi yang telah disepakati di awal serta memenuhi tujuannya. Selain itu, tahap pengujian adalah tahap untuk melakukan debugging dimana bug dan gangguan sistem dapat ditemukan, diperbaiki, dan hasil perangkat lunak dapat disempurnakan.

Pengujian perangkat lunak berisi kegiatan yang bertujuan untuk mencari dan mengevaluasi kesalahan atau *error* yang dapat ditemukan pada suatu perangkat lunak. Sasaran dari *testing* adalah untuk mencari *error*, sehingga pengujian yang baik adalah pengujian yang memiliki peluang yang besar untuk menemukan sebuah *error* (Pressman, 2010).

2.10.1 Black-Box Testing

Semua produk hasil rekayasa dapat diuji melalui dua cara, salah satunya adalah dengan mengetahui fungsi tertentu yang dirancang untuk dapat dilakukan oleh suatu produk. Pendekatan yang melalui sudut pandang *external* ini disebut sebagai *black-box testing*. *Black-box testing* mengacu pada pengujian yang dilakukan di antarmuka dari perangkat lunak dan berfokus kepada kebutuhan fungsional dari perangkat lunak tersebut. *Black-box testing* berusaha untuk mencari kesalahan mengenai fungsi yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data atau akses basis data dari pihak eksternal, kesalahan perilaku atau kinerja, serta kesalahan inisialisasi dan terminasi. Dalam melakukan *black-box testing* terdapat beberapa metode yang dapat dilakukan, pada penelitian ini metode yang akan digunakan antara lain adalah *Validation Testing*.

2.10.1.1 Validation Testing

Validation Testing merupakan pengujian perangkat lunak yang berfokus kepada validasi aksi-aksi yang terlihat oleh pengguna dan keluaran dari sistem yang dikenali oleh pengguna (Pressman, 2010). Penjelasan sederhana dari validasi disini adalah pengesahan ketika fungsi-fungsi yang ada pada perangkat lunak dapat berfungsi sesuai dengan harapan pengguna. Validasi perangkat lunak dapat dilakukan melalui serangkaian percobaan, tujuannya adalah untuk menunjukkan kesesuaiannya dengan kebutuhan sistem yang sudah dijelaskan di awal. Diagram *use case* dapat digunakan untuk menyusun rangkaian percobaan dalam validasi perangkat lunak, karena diagram ini termasuk dalam bagian dari pemodelan kebutuhan. *Use case* digunakan karena diagram ini menyediakan skenario yang berpeluang untuk menemukan kesalahan dalam kebutuhan interaksi pengguna.

2.10.2 Usability Testing

Usability Testing merupakan pengujian yang melibatkan perwakilan pengguna dari perangkat lunak dengan mencoba melakukan *task-task* perwakilan pada antarmuka purwarupa tahap awal, yang bertujuan untuk meningkatkan kualitas antarmuka dengan cara mencari kekurangan dari antarmuka yang perlu dilakukan perbaikan (Lazar, et al., 2017). Antarmuka yang diuji umumnya berupa tata letak untuk *desktop*, *laptop*, atau *handphone* dan beberapa macam perangkat bergerak lainnya. Tetapi pengujian *usability* juga dapat dilakukan untuk menguji interaksi fisik antara pengguna dengan perangkat dan sistem. Dalam melakukan pengujian *usability*, terdapat beberapa metrik yang dapat digunakan untuk mengukur hasil dan mengetahui tingkat *usability* dari pengujian. Standar ISO/IEC 9126-4 merekomendasikan bahwa metrik dari *usability* harus mencakup perihal *effectiveness*, *efficiency* dan *satisfaction* (Mifsud, 2015).

2.10.2.1 Metrik Effectiveness Usability Testing

Metrik *Effectiveness* mencakup tentang tingkat akurasi dan kelengkapan yang didapat dari pengguna untuk menyelesaikan *task* tertentu (Mifsud, 2015). *Effectiveness* dapat dihitung dengan mengukur tingkat penyelesaian. Tingkat penyelesaian dihitung dengan menggunakan nilai biner '1' jika responden berhasil menyelesaikan suatu *task* dan '0' jika tidak. *Effectiveness* dapat direpresentasikan menggunakan persentase dengan menggunakan Rumus 2.1 berikut:

$$\text{Effectiveness} = \frac{\text{Jumlah task yang berhasil diselesaikan}}{\text{Total jumlah task yang dilakukan}} \times 100\% \quad (2.1)$$

2.10.2.2 Metrik *Efficiency Usability Testing*

Metrik *Efficiency* mencakup tentang sumber daya waktu, baik dalam satuan detik maupun menit, yang dibutuhkan sehubungan dengan keakuratan dan kelengkapan dari pengguna untuk mencapai menyelesaikan *task* tertentu (Mifsud, 2015). *Efficiency* dapat dihitung dengan mengukur *Time-Based Efficiency* (TBE). *Time-Based Efficiency* dapat direpresentasikan menggunakan satuan *goals/sec* yang didapatkan dengan menggunakan Rumus 2.2 berikut:

$$TBE = \frac{\sum_{j=1}^R \sum_{i=1}^N \frac{n_{ij}}{t_{ij}}}{N \times R} \quad (2.2)$$

Keterangan:

N = Total jumlah dari *task*

R = Jumlah pengguna

n_{ij} = Hasil dari *task* yang dilakukan oleh pengguna j ; jika pengguna berhasil maka nilainya adalah 1, jika tidak berhasil maka nilainya adalah 0

t_{ij} = Waktu yang dibutuhkan oleh pengguna j untuk menyelesaikan *task* ke i ; jika *task* tidak berhasil diselesaikan maka waktu yang diukur adalah sampai pengguna tidak mampu menyelesaikan.

2.10.2.3 Metrik *Satisfaction Usability Testing*

Metrik *user satisfaction* atau kepuasan pengguna mencakup kenyamanan dan penerimaan dari pengguna untuk menggunakan suatu perangkat atau sistem. Metrik *satisfaction* dapat diukur dengan menggunakan *standardized satisfaction questionnaires* atau kuesioner kepuasan yang terstandarisasi yang dapat diberikan setelah setiap *task* selesai dilakukan atau setelah sesi pengujian berakhir (Mifsud, 2015).

Pada penelitian ini akan digunakan kuesioner *task level satisfaction* atau yang diberikan setelah setiap *task* selesai dilakukan. Kuesioner *task level satisfaction* yang diberikan berupa pertanyaan *Single Ease Question* (SEQ), yaitu berupa pertanyaan yang menanyakan kepada pengguna mengenai seberapa sulit atau mudah untuk menyelesaikan *task* yang telah diberikan dan dijawab dengan menggunakan poin-poin skala *likert* (Sauro, 2012).

2.10.2.4 Skala *Likert*

Skala *Likert* merupakan skala tanggapan psikometri yang sering digunakan dalam kuesioner untuk mendapatkan tingkat persetujuan dari responden menggunakan pernyataan atau serangkaian pernyataan (Bertram, 2016). Umumnya *skala likert* menggunakan skala yang terdiri dari 5 poin secara terurut dimulai dari "Sangat Tidak Setuju" di salah satu ujung kemudian "Sangat Setuju" di ujung lainnya kemudian "Antara Setuju maupun Tidak Setuju" di tengah-tengahnya. Terkadang juga digunakan skala 4 poin atau angka genap lainnya yang

bertujuan untuk menghasilkan ukuran ipsatif atau pilihan paksa. Setiap tingkat pada skala diberikan nilai numerik yang biasanya dimulai dari 1 dan ditambah satu untuk tiap tingkatnya. Berikut pada Gambar 2.9 merupakan contoh dari skala *likert* dengan menggunakan skala 5 poin.



Gambar 2.9 Contoh Skala Likert

Sumber : Bertram (2016)

Dalam implementasi skala *likert* terdapat beberapa pernyataan yang masing-masing memiliki bobot nilai tertentu. Proses perhitungan hasil dari skala *likert* diawali dengan memberikan bobot nilai pada tiap jawaban. Berikut pada Tabel 2.6 merupakan rincian bobot nilai dari jawaban. Sedangkan pada Tabel 2.7 merupakan persentase untuk setiap jawaban.

Tabel 2.6 Bobot Nilai Jawaban

Pilihan Jawaban	Bobot Nilai
A = Sangat Setuju	5
B = Setuju	4
C = Netral	4
D = Kurang	2
E = Sangat Kurang	1

Sumber: Choizes (2017)

Tabel 2.7 Persentase Nilai Jawaban

Interval Persentase	Interpretasi Skor
80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Kurang Setuju

Sumber: Choizes (2017)

Setelah bobot nilai dan persentase nilai telah ditentukan, selanjutnya adalah mengkalikan jumlah jawaban dengan bobot nilainya. Untuk perhitungannya akan dijelaskan pada Rumus 2.3 berikut:

1. Jawaban Sangat Setuju (A) = $n \times 5$
2. Jawaban Setuju (B) = $n \times 4$
3. Jawaban Netral (C) = $n \times 3$
4. Jawaban Tidak Setuju (D) = $n \times 2$
5. Jawaban Sangat Tidak Setuju (E) = $n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.3)$$

n merupakan jumlah dari responden yang menjawab. Perhitungan selanjutnya adalah mencari nilai terendah (Y) dan nilai tertinggi dari hasil skala *likert* tersebut. Rumus yang digunakan dijelaskan pada Rumus 2.4 dan Rumus 2.5, dimana $N1$ merupakan nilai tertinggi skala *likert*, $N2$ merupakan nilai terendah skala *likert*, n merupakan jumlah responden, dan U merupakan jumlah uji kasus.

$$Y = N1 \times n \times U \quad (2.4)$$

$$Y = N2 \times n \times U \quad (2.5)$$

Sedangkan untuk menghitung hasil dari UAT dengan menggunakan skala *likert* dijelaskan pada Rumus 2.6 berikut.

$$\text{Rumus Index \%} = (\text{Total Nilai} \div Y) \times 100 \quad (2.6)$$

2.11 Tahap Pemeliharaan

Tahap Pemeliharaan atau *Maintenance Phase* merupakan proses memodifikasi perangkat lunak setelah pengiriman dan penerapan untuk memperbaiki output, memperbaiki kesalahan, dan meningkatkan kinerja dan kualitas. Tahap ini termasuk mengadaptasi perangkat lunak ke lingkungannya, mengakomodasi kebutuhan pengguna baru, dan meningkatkan keandalan perangkat lunak.

Akan tetapi dalam penelitian untuk pengembangan Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana ini, proses pengembangan hanya akan dilakukan mulai dari Tahap Analisis hingga Tahap Pengujian saja. Tahap Pemeliharaan tidak akan dibahas lebih lanjut dalam penelitian ini.

BAB 3 METODOLOGI

3.1 Metodologi Penelitian

Metode penelitian yang akan dilakukan terdiri dari beberapa proses yang dilakukan secara berurutan, berikut merupakan diagram alir dari penelitian ini yang ditunjukkan pada Gambar 3.1 berikut.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1.1 Wawancara

Pada tahap ini, peneliti melakukan pengumpulan data dengan menggunakan teknik wawancara. Penelitian ini merupakan bagian dari pengembangan sistem informasi yang diharapkan dapat diterapkan di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya untuk membantu civitas akademika dalam menyampaikan keluhan mengenai sarana prasarana yang ada di lingkup area FILKOM. Wawancara awal yang terdapat pada Bab Lampiran A Hasil Wawancara Pegawai Perlengkapan, dilakukan terhadap beberapa pegawai perlengkapan di Sub Bagian Umum dan Perlengkapan FILKOM guna mengetahui alur proses bisnis, *stakeholder* yang

terlibat, serta permasalahan-permasalahan yang ada di proses penanganan laporan keluhan sarana prasarana yang ada di FILKOM. Sedangkan wawancara tahap berikutnya yang terdapat pada Bab Lampiran B Hasil Wawancara Mahasiswa Civitas FILKOM UB, dilakukan terhadap beberapa mahasiswa FILKOM sebagai perwakilan pihak pelapor dari civitas akademik FILKOM UB guna mengetahui proses pelaporan keluhan sarana prasarana yang dilakukan beserta dengan masalah-masalah yang muncul dalam melaporkan keluhan tersebut selama ini. Hasil dari wawancara ini nantinya akan digunakan untuk menganalisis dan menentukan kebutuhan dari sistem yang akan dikembangkan.

3.1.2 Studi Pustaka

Pada tahap studi pustaka ini, peneliti melakukan pencarian dan pengumpulan referensi dari jurnal penelitian, naskah penelitian, buku, *e-book* dan internet untuk mendapatkan penjelasan teori-teori yang dijadikan dasar untuk mendukung penelitian. Adapun studi literatur yang dimaksud berkaitan dengan hal-hal berikut:

1. Penelitian sejenis yang pernah dilakukan
2. Manajemen Aset
3. Sistem Informasi Manajemen Aset
4. Sistem Informasi
5. Business Process Model & Notation
6. Software Development Life Cycle
7. Waterfall Model
8. Unified Modelling Language (UML)
9. Pengujian Perangkat Lunak

3.1.3 Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem merupakan penerapan dari Tahap Analisis dan merupakan tahap pertama dari SDLC menggunakan *Waterfall Model*. Data yang didapatkan dari hasil wawancara yang dilakukan pada tahap Wawancara digunakan sebagai dasar analisis kebutuhan sistem yang akan dikembangkan. Pada tahap ini akan dilakukan pemodelan proses bisnis yang berjalan, analisis proses bisnis yang berjalan, pemodelan proses bisnis setelah adanya sistem, analisis proses bisnis sistem setelah adanya sistem, analisis kebutuhan dari *stakeholder* dan pemodelan kebutuhan dari sistem dengan menggunakan diagram *use case* dan *use case scenario*.

3.1.4 Perancangan Sistem

Pada tahap ini akan dilakukan perancangan sistem berdasarkan hasil analisis kebutuhan fungsional dan non fungsional yang didapatkan dari proses analisis dan pemodelan kebutuhan system sebelumnya. Perancangan sistem sendiri merupakan penerapan dari Tahap Perancangan dari SDLC menggunakan *Waterfall Model*. Perancangan sistem akan direpresentasikan menggunakan diagram UML seperti diagram sekuen dan diagram kelas. Langkah selanjutnya adalah membuat rancangan *user interface*(UI) atau tampilan yang akan digunakan dalam sistem. Rancangan UI nantinya akan menggambarkan tata letak dari tombol, fungsi dan penampilan data yang ada pada sistem.

3.1.5 Implementasi Sistem

Tahap implementasi sistem merupakan proses pengerjaan Sistem Informasi Manajemen Aset Pelaporan dan Pemeliharaan Sarana Prasarana berdasarkan perancangan yang telah dibuat sebelumnya. Tahapan implementasi sistem ini merupakan penerapan dari Tahap Implementasi dari SDLC menggunakan *Waterfall Model*. Pada tahap ini dilakukan penulisan kode-kode sebagai implementasi dari rancangan sistem yang telah dirancang pada tahap sebelumnya dimana sistem dikembangkan menggunakan bahasa pemrograman *Java* untuk mengembangkan aplikasi *mobile* berbasis *Android* yang digunakan oleh pelapor dan bahasa pemrograman HTML, PHP dan Javascript untuk mengembangkan *website operator* yang digunakan oleh Pegawai Perlengkapan sebagai *operator*.

3.1.6 Pengujian Sistem

Pada tahap ini dilakukan pengujian dan analisis dari sistem yang berarti merupakan penerapan dari Tahap Pengujian dari SDLC menggunakan *Waterfall Model*. Pengujian dilakukan untuk mengetahui tingkat keberhasilan sistem ketika digunakan apakah telah sesuai dengan perancangan awal atau tidak dan untuk mengetahui terpenuhi atau tidaknya ekspektasi dari *stakeholder*. Pengujian dilakukan dengan menggunakan *black-box testing* yang terdiri dari *validation testing* dan *user acceptance testing*. *Validation testing* dilakukan untuk melakukan validasi dari aksi-aksi sistem yang terlihat oleh pengguna dan keluaran dari sistem yang dikenali oleh pengguna. Sedangkan *user acceptance testing* berfokus pada pengujian sistem untuk menentukan apakah sistem yang dibangun telah memenuhi kriteria untuk diterima oleh pengguna atau instansi.

3.1.7 Kesimpulan dan Saran

Tahap terakhir dari penelitian ini adalah penarikan kesimpulan dari keseluruhan hasil penelitian dan pemberian saran yang dapat dijadikan sebagai acuan untuk melakukan pengembangan aplikasi oleh penelitian selanjutnya.

BAB 4 ANALISIS KEBUTUHAN

Analisis Kebutuhan merupakan representasi dari Tahap Analisis pada *Waterfall Model*, yang mana merupakan proses pengembangan yang melibatkan proses analisis kebutuhan. Analisis kebutuhan menggambarkan, menyebutkan dan menjelaskan kebutuhan-kebutuhan dari perangkat lunak yang harus dimiliki dan dapat dilakukan oleh sistem agar dapat memenuhi kebutuhan pengguna. Tahapan yang dilakukan pada analisis kebutuhan terdiri dari dua tahap, yaitu tahap rekayasa kebutuhan dan pemodelan kebutuhan.

4.1 Rekayasa Kebutuhan

Tahap rekayasa kebutuhan merupakan tahap untuk mengumpulkan dan mendefinisikan kebutuhan-kebutuhan dari sistem yang akan dibangun. Tahap ini terdiri dari beberapa proses yang mana antara lain adalah analisis proses bisnis *as-is*, elisitasi kebutuhan, identifikasi aktor, pemodelan proses bisnis *to-be* dan spesifikasi kebutuhan.

Analisis proses bisnis *as-is* merupakan proses melakukan analisis dari permasalahan yang ada kemudian dilakukan pemodelan menggunakan *Business Process Modelling Notation* (BPMN). Selanjutnya dalam melakukan elisitasi kebutuhan pada proses pendefinisian kebutuhan sistem, teknik dan langkah yang digunakan adalah melalui wawancara dengan dua pihak *stakeholder* seperti yang terdapat pada Bab Lampiran A Hasil Wawancara Pegawai Perlengkapan serta yang terdapat pada Bab Lampiran B Hasil Wawancara Mahasiswa Civitas FILKOM UB. Kemudian setelah didapatkan permasalahan dan kebutuhan untuk perbaikan dari sistem yang ada, akan dilanjutkan dengan identifikasi aktor agar dapat digambarkan proses bisnis yang akan diajukan berupa proses bisnis *to-be* sehingga proses selanjutnya yaitu spesifikasi kebutuhan dapat dilakukan.

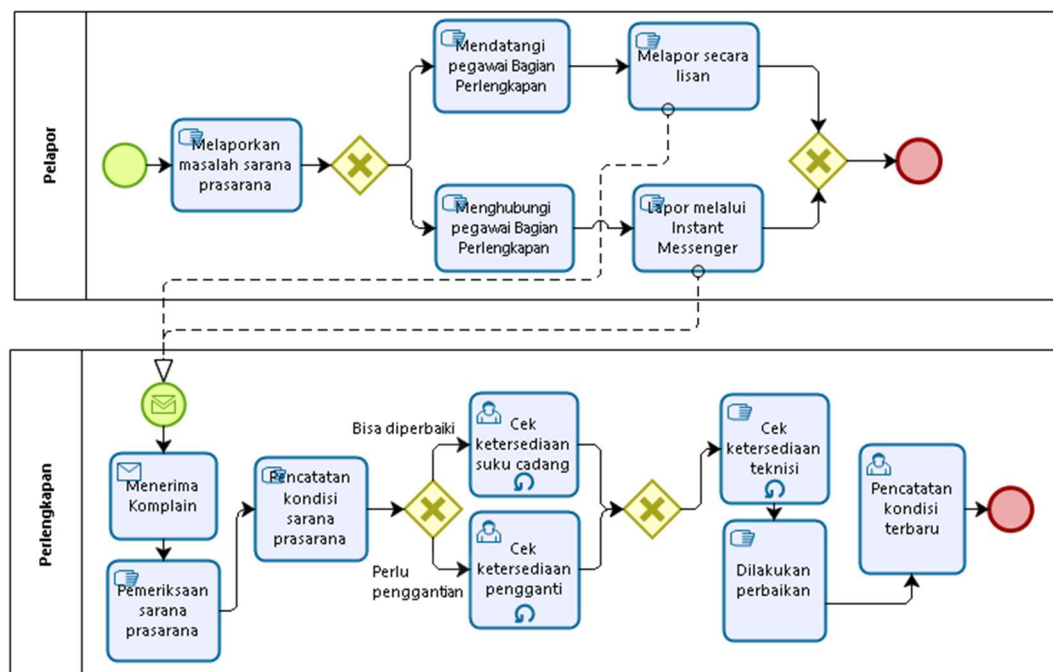
4.1.1 Analisis Proses Bisnis As-Is

Analisis dari sudut pandang bisnis akan dilakukan analisis proses bisnis melalui pemodelan menggunakan BPMN. BPMN digunakan untuk menggambarkan proses bisnis saat ini dan yang akan diusulkan pada Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Analisis proses bisnis *as-is* pada Tahap Analisis ini dilakukan untuk mengidentifikasi proses bisnis tentang pelaporan keluhan mengenai masalah sarana prasarana oleh seluruh civitas akademik yang berada di FILKOM Universitas Brawijaya. Proses bisnis yang sedang berjalan saat ini akan digambarkan dalam diagram BPMN yang setelah itu akan dilakukan analisis berdasarkan masalah yang ada. Setelah analisis berdasarkan masalah yang ada telah dilakukan, maka solusi perbaikan atau perubahan dari permasalahan tersebut bisa didapatkan dan akan digambarkan dalam proses bisnis *to-be*. Berdasarkan hasil wawancara yang terdapat pada Bab Lampiran A Hasil Wawancara Pegawai Perlengkapan yang dilakukan dengan pegawai Sub Bagian Perlengkapan di FILKOM, didapatkan hasil bahwa proses bisnis *as-is* untuk pelaporan keluhan mengenai sarana prasarana dapat dibagi menjadi 2 jenis proses

bisnis yang berbeda. Adapun 2 jenis proses bisnis tersebut yang pertama adalah proses bisnis pelaporan keluhan yang disampaikan secara langsung oleh pihak pelapor kepada pegawai Perlengkapan, dan yang kedua adalah pelaporan keluhan yang disampaikan oleh pihak pelapor melalui E-Complaint UB.

4.1.1.1 Proses Pelaporan Keluhan Sarana Prasarana secara Langsung

Untuk proses bisnis pelaporan keluhan yang disampaikan secara langsung kepada pegawai Perlengkapan, terdapat 2 aktor yang terlibat yaitu antara lain adalah pihak Pelapor yang merupakan seluruh civitas akademik FILKOM dan pegawai dari bagian Perlengkapan. Penggambaran proses bisnis *as-is* untuk pelaporan keluhan mengenai sarana prasarana secara langsung kepada pegawai Perlengkapan ditunjukkan pada Gambar 4.1:



Gambar 4.1 Proses Bisnis As-Is Pelaporan Keluhan Sarana Prasarana Secara Langsung

Penjelasan mengenai proses bisnis *as-is* untuk pelaporan keluhan mengenai sarana prasarana secara langsung pada Gambar 4.1 antara lain adalah sebagai berikut:

1. Pelapor melaporkan masalah mengenai sarana prasarana yang berada di lingkup wilayah FILKOM UB.
2. Pelapor menyampaikan keluhan kepada pegawai Bagian Perlengkapan. Dalam menyampaikan keluhan, pelapor dapat melaporkannya secara lisan dengan menemui pegawai Bagian Perlengkapan secara langsung atau dengan melaporkannya melalui media telepon, SMS atau *Instant Messenger*.

3. Pelapor menjelaskan mengenai barang apa yang bermasalah, lokasi dari barang yang bermasalah dan permasalahan yang terjadi.
4. Pegawai Bagian Perlengkapan menerima keluhan yang disampaikan secara lisan atau melalui media telepon, SMS atau Instant Messenger.
5. Pegawai Bagian Perlengkapan melakukan pemeriksaan sarana prasarana yang telah diterima keluhannya untuk mengetahui permasalahan yang terjadi.
6. Pegawai Bagian Perlengkapan melakukan pencatatan kondisi dari sarana prasarana yang sedang diperiksa.
7. Kemudian pegawai Bagian Perlengkapan menentukan apakah sarana prasarana yang bermasalah tersebut dapat diperbaiki hanya dengan melakukan penggantian suku cadang ataukah harus dilakukan penggantian dengan barang baru.
8. Apabila sarana prasarana tersebut dapat diperbaiki dengan melakukan penggantian suku cadang, maka pegawai Bagian Perlengkapan perlu memeriksa ketersediaan suku cadang pengganti dari sarana prasarana tersebut. Jika suku cadang tersedia maka perbaikan dapat dilakukan, jika belum tersedia maka perlu menunggu datangnya suku cadang.
9. Sedangkan apabila sarana prasarana tersebut perlu dilakukan penggantian dengan barang baru, maka pegawai Bagian Perlengkapan perlu memeriksa ketersediaan barang pengganti dari sarana prasarana tersebut. Jika barang pengganti tersedia maka penggantian dapat segera dilakukan, jika belum tersedia maka perlu menunggu datangnya barang pengganti.
10. Setelah itu pegawai Bagian Perlengkapan memastikan apakah teknisi untuk melakukan perbaikan atau penggantian barang sarana prasarana tersebut tersedia atau tidak. Jika teknisi telah tersedia maka perbaikan atau penggantian dapat segera dilaksanakan, sedangkan jika teknisi belum tersedia maka perbaikan atau penggantian perlu menunggu tersedianya teknisi.
11. Setelah perbaikan atau penggantian selesai dilaksanakan, kemudian pegawai Bagian Perlengkapan perlu melakukan pencatatan status dan kondisi terkini dari sarana prasarana yang telah diperbaiki atau diganti.

Berdasarkan proses bisnis as-is untuk pelaporan keluhan mengenai sarana prasarana secara langsung pada Gambar 4.1 tersebut, didapatkan aktifitas-aktifitas yang dikerjakan beserta aktor yang mengerjakan, waktu penyelesaian terbaiknya, penyelesaian terburuk dan waktu rata-rata penyelesaiannya pada Tabel 4.1 berikut.

Tabel 4.1 Penyelesaian Waktu Aktivitas Pelaporan As-Is Secara Langsung

No.	Aktor	Aktivitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pelapor	Melaporkan masalah sarana prasarana	30 detik	2 menit	1 menit
2	Pelapor	Mendatangi Pegawai Perlengkapan	3 menit	8 menit	5 menit
3	Pelapor	Melapor secara lisan	1 menit	3 menit	2 menit
4	Pelapor	Menghubungi pegawai Perlengkapan	10 detik	1 menit	30 detik
5	Pelapor	Melapor melalui Instant Messenger	30 detik	2 menit	1 menit
6	Perlengkapan	Menerima keluhan	10 menit	30 menit	20 menit
7	Perlengkapan	Pemeriksaan sarana prasarana	30 menit	2 jam	75 menit
8	Perlengkapan	Pencatatan kondisi sarana prasarana	30 menit	1 jam	45 menit
9	Perlengkapan	Cek ketersediaan suku cadang/pengganti	10 menit	30 menit	20 menit
10	Perlengkapan	Cek ketersediaan teknisi	10 menit	30 menit	20 menit
11	Perlengkapan	Dilakukan perbaikan	1 jam	3 hari	2 jam
12	Perlengkapan	Pencatatan kondisi terbaru	10 menit	20 menit	15 menit
Total Rata-Rata Waktu			2 jam 45 menit 10 detik	3 hari 4 jam 1 menit	5 jam 24 menit 30 detik

Berdasarkan Tabel 4.1 tersebut diketahui bahwa untuk aktifitas pelaporan dari proses bisnis *as-is* secara langsung didapatkan waktu penyelesaian terbaiknya memerlukan waktu 2 jam 45 menit 10 detik, waktu penyelesaian terburuknya memerlukan 3 hari 4 jam 1 menit dan rata-rata dari waktu penyelesaiannya memerlukan 5 jam 24 menit 30 detik. Terjadinya perbedaan waktu dari waktu penyelesaian terbaik dan terburuk dari proses bisnis *as-is* tersebut diakibatkan oleh beberapa faktor-faktor.

Pada Tabel 4.1 tersebut didapatkan bahwa apabila civitas akademik ingin melaporkan keluhan sarana prasarana, maka mereka harus melakukan aktivitas Mendatangi Pegawai Perlengkapan yang mana membutuhkan waktu 3 menit sampai 8 menit karena lokasi dari ruangan Pegawai Perlengkapan berada di Lantai

6 Gedung F FILKOM. Pelaporan juga dapat dilakukan dengan aktifitas Menghubungi Pegawai Perlengkapan terlebih dahulu yang memiliki waktu proses lebih singkat. Tetapi berdasarkan hasil wawancara yang dilakukan dengan civitas akademik khususnya dari mahasiswa, mereka masih segan untuk menghubungi Pegawai Perlengkapan untuk melaporkan keluhan yang dialami.

Sedangkan untuk waktu aktifitas pelacakan status pelaporan secara lisan dan melalui *Instant Messenger* dapat dilihat pada Tabel 4.2. Berdasarkan Tabel 4.2 tersebut, diketahui bahwa untuk mendapatkan status pelaporan dengan cara menanyakan secara lisan bagi pelapor harus mendatangi ruangan Pegawai Perlengkapan terlebih dahulu yang membutuhkan waktu 3 menit sampai 8 menit. Sedangkan jika pelapor menanyakan dengan menghubungi melalui *Instant Messenger* membutuhkan waktu 1 menit sampai dengan 1 hari kerja. Hal ini disebabkan karena kesibukan yang dialami oleh Pegawai Perlengkapan sehingga mereka tidak dapat cepat dalam membalas setiap pertanyaan mengenai status pelaporan dari masing-masing pihak pelapor.

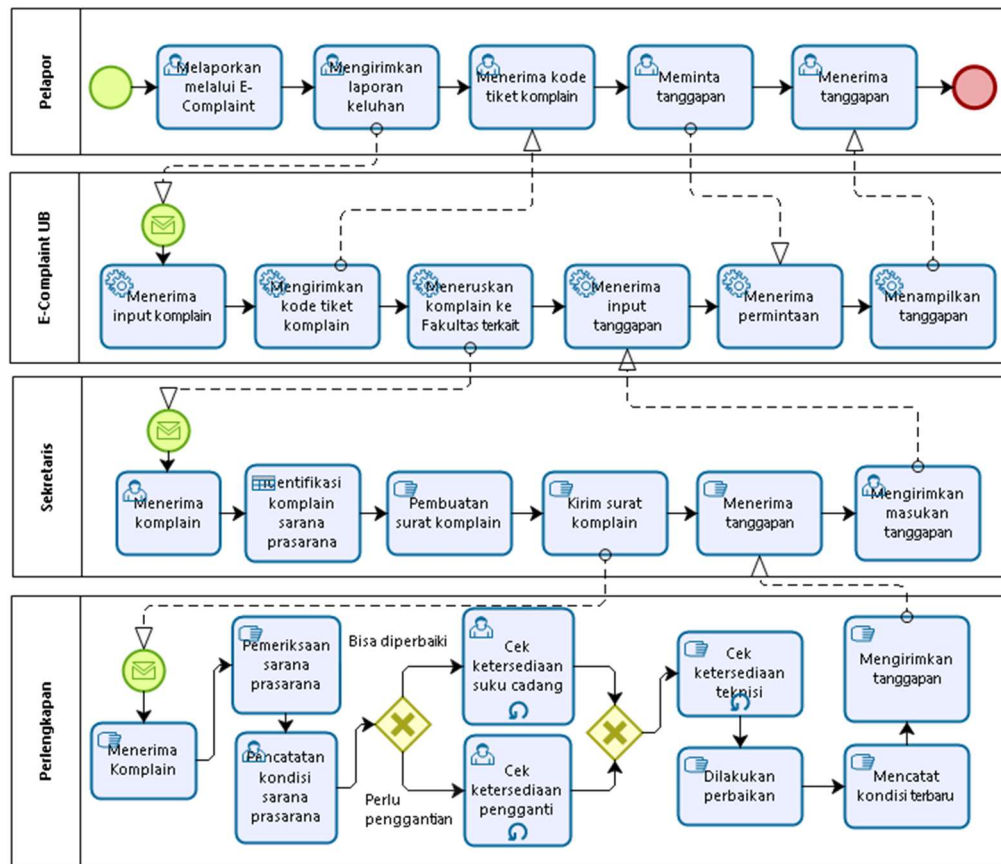
Tabel 4.2 Penyelesaian Waktu Aktifitas Pelacakan Status Pelaporan As-Is secara lisan dan *Instant Messenger*

No.	Aktor	Aktifitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pelapor	Mendatangi Pegawai Perlengkapan	3 menit	8 menit	5 menit
2	Pelapor	Menerima tanggapan secara lisan	10 detik	1 menit	30 detik
3	Pelapor	Menghubungi pegawai Perlengkapan	10 detik	1 menit	30 detik
4	Pelapor	Menerima tanggapan melalui <i>Instant Messenger</i>	1 menit	1 hari	6 jam
Total Rata-Rata Waktu			4 menit 20 detik	1 hari 10 menit	6 jam 6 menit

4.1.1.2 Proses Pelaporan Keluhan Sarana Prasarana melalui E-Complaint UB

Berikutnya adalah proses bisnis untuk pelaporan keluhan yang disampaikan melalui E-Complaint UB. Pada proses bisnis ini terdiri dari 4 aktor yang terlibat yaitu antara lain adalah pihak Pelapor yang merupakan seluruh civitas akademik FILKOM, sistem E-Complaint UB sebagai sarana untuk melaporkan keluhan, pegawai Sekretaris sebagai pengelola dari sistem E-Complaint UB untuk FILKOM dan yang terakhir adalah pegawai dari bagian Perlengkapan. Penggambaran

proses bisnis *as-is* untuk pelaporan keluhan mengenai sarana prasarana melalui E-Complaint UB ditunjukkan pada Gambar 4.2:



Gambar 4.2 Proses Bisnis *As-Is* Pelaporan Keluhan Sarana Prasarana Melalui E-Complaint UB

Penjelasan mengenai proses bisnis *as-is* untuk pelaporan keluhan mengenai sarana prasarana melalui E-Complaint UB pada Gambar 4.2 antara lain adalah sebagai berikut:

1. Pelapor melaporkan masalah mengenai sarana prasarana yang berada di lingkup wilayah FILKOM UB melalui website E-Complaint UB.
2. Pelapor menyampaikan keluhan dengan cara mengisi dan mengirimkan data-data masukan sesuai dengan *formulir* yang perlu diisi pada website E-Complaint UB.
3. Data keluhan yang telah dimasukkan tersebut disimpan ke dalam basis data dari website E-Complaint UB yang nantinya data tersebut dapat diakses oleh pengelola dari masing-masing unit kerja/fakultas.
4. Pelapor mendapatkan kode tiket keluhan dari website E-Complaint UB untuk melacak status dan tanggapan keluhan.
5. Kemudian pegawai Bagian Sekretaris mengidentifikasi keluhan yang masuk kemudian dikelompokkan berdasarkan lingkup keluhannya.

6. Pegawai Bagian Sekretaris mengumpulkan keluhan yang berisi keluhan mengenai sarana prasarana lalu membuat surat yang ditujukan kepada pegawai Bagian Perlengkapan.
7. Surat keluhan diserahkan kepada pegawai Bagian Perlengkapan untuk segera ditindak lanjuti.
8. Pegawai Bagian Perlengkapan menerima surat keluhan dari Bagian Sekretaris.
9. Pegawai Bagian Perlengkapan melakukan pemeriksaan sarana prasarana yang telah diterima keluhannya untuk mengetahui permasalahan yang terjadi.
10. Pegawai Bagian Perlengkapan melakukan pencatatan kondisi dari sarana prasarana yang sedang diperiksa.
11. Pegawai Bagian Perlengkapan menentukan apakah sarana prasarana yang bermasalah tersebut dapat diperbaiki hanya dengan melakukan penggantian suku cadang ataukah harus dilakukan penggantian dengan barang baru.
12. Apabila sarana prasarana tersebut dapat diperbaiki dengan melakukan penggantian suku cadang, maka pegawai Bagian Perlengkapan perlu memeriksa ketersediaan suku cadang pengganti dari sarana prasarana tersebut. Jika suku cadang tersedia maka perbaikan dapat dilakukan, jika belum tersedia maka perlu menunggu datangnya suku cadang.
13. Sedangkan apabila sarana prasarana tersebut perlu dilakukan penggantian dengan barang baru, maka pegawai Bagian Perlengkapan perlu memeriksa ketersediaan barang pengganti dari sarana prasarana tersebut. Jika barang pengganti tersedia maka penggantian dapat segera dilakukan, jika belum tersedia maka perlu menunggu datangnya barang pengganti.
14. Kemudian pegawai Bagian Perlengkapan memastikan ketersediaan teknisi untuk melakukan perbaikan atau penggantian barang sarana prasarana. Jika teknisi telah tersedia maka perbaikan atau penggantian dapat segera dilaksanakan, sedangkan jika teknisi belum tersedia maka perbaikan atau penggantian perlu menunggu tersedianya teknisi.
15. Setelah perbaikan atau penggantian selesai dilaksanakan, kemudian pegawai Bagian Perlengkapan melakukan pencatatan status dan kondisi terkini dari sarana prasarana yang telah diperbaiki atau diganti.
16. Pegawai Bagian Perlengkapan membuat surat tanggapan dari keluhan yang telah ditindak lanjuti dan menyerahkannya kepada pegawai Bagian Sekretaris.
17. Pegawai Bagian Sekretaris memasukkan tanggapan dari Pegawai Bagian Perlengkapan kedalam website E-Complaint UB.

18. E-Complaint UB menerima dan menyimpan tanggapan yang telah dimasukkan.
19. Pelapor dapat memeriksa status tanggapan mengenai keluhan yang telah dilaporkan melalui akun pada E-Complaint UB dengan memasukkan kode tiket yang telah didapatkan.
20. E-Complaint UB menerima permintaan tanggapan lalu menampilkan tanggapan kepada pelapor.

Berdasarkan proses bisnis as-is untuk pelaporan keluhan mengenai sarana prasarana melalui E-Complaint UB pada Gambar 4.2 tersebut, didapatkan aktifitas-aktifitas yang dikerjakan beserta aktor yang mengerjakan, waktu penyelesaian terbaiknya, penyelesaian terburuk dan waktu rata-rata penyelesaiannya pada Tabel 4.1 berikut.

Tabel 4.3 Penyelesaian Waktu Aktifitas Pelaporan As-Is melalui E-Complaint

No.	Aktor	Aktifitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pelapor	Melaporkan melalui E-Complaint	4 menit	6 menit	5 menit
2	Pelapor	Mengirimkan laporan keluhan	5 detik	10 detik	7 detik
3	E-Complaint UB	Menerima masukan keluhan	1 detik	3 detik	2 detik
4	E-Complaint UB	Mengirimkan kode tiket keluhan	1 detik	3 detik	2 detik
5	E-Complaint UB	Meneruskan keluhan ke fakultas terkait	30 detik	1 menit	45 detik
6	Sekretaris	Menerima keluhan	20 menit	8 jam	4 jam
7	Sekretaris	Identifikasi keluhan sarana prasarana	1 jam	16 jam	8,5 jam
8	Sekretaris	Pembuatan surat keluhan	1 hari	3 hari	2 hari
9	Sekretaris	Kirim surat keluhan	10 menit	1 jam	35 menit
10	Perlengkapan	Menerima keluhan	10 menit	30 menit	20 menit
11	Perlengkapan	Pemeriksaan sarana prasarana	30 menit	2 jam	75 menit
12	Perlengkapan	Pencatatan kondisi sarana prasarana	30 menit	1 jam	45 menit

No.	Aktor	Aktifitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
13	Perlengkapan	Cek ketersediaan suku cadang/pengganti	10 menit	30 menit	20 menit
14	Perlengkapan	Cek ketersediaan teknisi	10 menit	30 menit	20 menit
15	Perlengkapan	Dilakukan perbaikan	1 jam	3 hari	2 jam
16	Perlengkapan	Mencatat kondisi terbaru	10 menit	20 menit	15 menit
17	Perlengkapan	Mengirimkan tanggapan	30 menit	3 jam	105 menit
18	Sekretaris	Menerima tanggapan	10 menit	30 menit	20 menit
19	Sekretaris	Mengirimkan masukan tanggapan	2 hari	3 hari	2,5 hari
20	E-Complaint UB	Menerima input tanggapan	1 detik	5 detik	3 detik
Total Rata-Rata Waktu			2 hari 2 jam 4 menit 38 detik	10 hari 9 jam 27 menit 22 detik	5 hari 8 jam 30 menit 59 detik

Berdasarkan Tabel 4.3 tersebut diketahui bahwa untuk aktifitas pelaporan dari proses bisnis *as-is* didapatkan waktu penyelesaian terbaiknya memerlukan waktu 2 hari 2 jam 4 menit 38 detik, waktu penyelesaian terburuknya memerlukan 10 hari 9 jam 27 menit 22 detik dan rata-rata dari waktu penyelesaiannya memerlukan 5 hari 8 jam 30 menit 59 detik. Terjadinya perbedaan waktu dari waktu penyelesaian terbaik dan terburuk dari proses bisnis *as-is* tersebut diakibatkan oleh beberapa faktor-faktor.

Faktor yang paling berpengaruh terhadap waktu penyelesaian terbaik dan terburuk adalah waktu dari aktifitas-aktifitas yang dilakukan oleh aktor Sekretaris dan Perlengkapan. Waktu penyelesaian dari aktifitas Identifikasi Keluhan Sarana Prasarana dan Pembuatan Surat Keluhan yang dilakukan oleh aktor Sekretaris bergantung kepada seberapa banyak keluhan yang diterima oleh Sekretaris dan bergantung kepada kehadiran dari Wakil Dekan yang memberikan persetujuan untuk proses identifikasi keluhan-keluhan yang masuk dan untuk disposisi terhadap masing-masing Bagian yang bertanggung jawab. Sedangkan waktu penyelesaian dari aktifitas Dilakukan Perbaikan yang dilakukan oleh aktor Perlengkapan, bergantung kepada bagaimana pengerjaan dari perbaikan yang dilakukan. Hal ini juga dipengaruhi oleh ketersediaan dari suku cadang atau barang

pengganti yang dibutuhkan untuk perbaikan yang dilakukan, serta ketersediaan dari teknisi yang mampu melakukan perbaikan dari kerusakan yang terjadi.

Berikut pada Tabel 4.4 menunjukkan hasil pengujian waktu untuk aktifitas-aktifitas pelacakan status pelaporan keluhan sarana prasarana yang mengacu kepada aktifitas pada proses bisnis *as-is*. Berdasarkan Tabel 4.4 tersebut diketahui bahwa untuk aktifitas pelacakan dari proses bisnis *as-is* didapatkan waktu penyelesaian terbaiknya memerlukan waktu 9 detik, waktu penyelesaian terburuknya memerlukan 19 detik dan rata-rata dari waktu penyelesaiannya memerlukan 14 detik. Perbedaan waktu ini disebabkan oleh aktifitas yang perlu dilakukan oleh pelapor untuk memasukkan kode tiket keluhan.

Tabel 4.4 Penyelesaian Waktu Aktifitas Pelacakan Status Pelaporan As-Is melalui E-Complaint

No.	Aktor	Aktifitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pelapor	Memasukkan kode tiket keluhan	6 detik	10 detik	8 detik
2	E-Complaint UB	Menerima permintaan tanggapan	1 detik	3 detik	2 detik
3	E-Complaint UB	Menampilkan tanggapan	1 detik	3 detik	2 detik
4	Pelapor	Menerima tanggapan	1 detik	3 detik	2 detik
Total Rata-Rata Waktu			9 detik	19 detik	14 detik

4.1.1.3 Analisis Permasalahan Saat Ini

Bagian ini merupakan penjelasan lebih rinci tentang permasalahan yang ditemukan dari hasil wawancara dan pemodelannya ke dalam BPMN. Penjelasan tersebut dipaparkan dalam bentuk tabel analisis permasalahan yang menjelaskan tentang masalah yang timbul, pihak yang terpengaruh masalah tersebut, dampak dari permasalahan dan solusi terhadap masalah. Berikut pada Tabel 4.5 merupakan analisis permasalahan proses bisnis pelaporan keluhan sarana prasarana secara langsung serta melalui E-Complaint UB.

Tabel 4.5 Analisis Permasalahan

Masalah	<p>Pelaporan Keluhan Secara Langsung :</p> <ol style="list-style-type: none"> 1. Waktu yang dibutuhkan untuk melaporkan secara lisan kepada Pegawai Perlengkapan cukup lama, karena jarak yang harus ditempuh oleh pelapor untuk mendatangi ruangan Pegawai Perlengkapan jauh. 2. Pihak pelapor dan civitas akademik lain kesusahan untuk mengetahui status pelaporan mengenai sarana prasarana yang telah disampaikan kepada Pegawai Perlengkapan. <p>Pelaporan Keluhan melalui E-Complaint UB:</p> <ol style="list-style-type: none"> 1. Proses <i>input</i> data dalam melaporkan keluhan mengenai sarana prasarana melalui <i>website E-Complaint</i> membutuhkan waktu 4-6 menit. Hal ini disebabkan karena pada <i>website E-Complaint</i> tidak tersedia layanan untuk mempermudah dan mempercepat identifikasi barang sarana prasarana yang bermasalah. 2. Proses pelacakan status pelaporan pada <i>website E-Complaint</i> mengharuskan pelapor untuk memasukkan kode tiket keluhan terlebih dahulu. Selain itu status pelaporan keluhan hanya dapat diketahui oleh pelapor dari keluhan tersebut sehingga berpotensi untuk terjadi pelaporan yang berulang oleh civitas lain. 3. Waktu yang dibutuhkan untuk diterimanya laporan mengenai sarana prasarana yang masuk melalui sistem E-Complaint oleh Bagian Perlengkapan mencapai 32-90 jam kerja.
Mempengaruhi	<ul style="list-style-type: none"> - Civitas Akademik FILKOM terdiri dari Mahasiswa, Dosen dan Karyawan dalam melaporkan keluhan sarana prasarana - Pegawai Bagian Perlengkapan FILKOM UB dalam menerima dan menindaklanjuti keluhan sarana prasarana
Dampak	<ol style="list-style-type: none"> 1. Pihak civitas akademik FILKOM yang melaporkan tidak dapat mengetahui status tindak lanjut dari keluhan yang dilaporkan. 2. Pihak civitas akademik FILKOM lain tidak mengetahui terdapat keluhan mengenai sarana prasarana apa saja yang sedang terjadi dan status penindak-lanjutannya.

	3. Tidak transparannya proses pelaporan keluhan sarana prasarana menimbulkan pihak pelapor enggan menyampaikan keluhan langsung kepada pegawai Bagian Perlengkapan.
Solusi	<ol style="list-style-type: none"> 1. Membangun sistem yang dapat digunakan oleh civitas akademik FILKOM untuk melaporkan keluhan mengenai sarana prasarana yang dapat menampilkan status tindak lanjut keluhan baik kepada pelapor maupun kepada civitas akademik lainnya. 2. Membangun sistem yang menyediakan fitur untuk mempermudah dan mempercepat identifikasi barang sarana prasarana yang bermasalah. 3. Membangun sistem yang dapat menyampaikan laporan keluhan yang langsung diterima oleh pegawai Bagian Perlengkapan.

4.1.2 Elisitasi Kebutuhan

Pada penelitian ini, elisitasi kebutuhan dilakukan dengan teknik wawancara sehingga pengembang dapat berinteraksi secara langsung dengan cara bertanya dan mendengar dari *stakeholder-stakeholder* sistem agar dapat merumuskan kebutuhan dari sistem. Wawancara dilakukan dengan dua pihak, yang pertama adalah dengan pihak pelapor dan yang kedua adalah dengan pihak *operator*. Pihak pelapor merupakan seluruh civitas akademik FILKOM yang mana pada penelitian ini pihak yang diwawancara diwakilkan dari pihak mahasiswa. Sedangkan pihak *operator* yang diwawancarai merupakan pegawai Bagian Perlengkapan sendiri yang diwakilkan oleh Ferix Panji Andrianto selaku pegawai Bagian Perlengkapan untuk pemeliharaan. Dari hasil dan analisis wawancara yang dilakukan terhadap pihak pelapor, didapatkan temuan kebutuhan sistem untuk pelapor yang dijelaskan pada Tabel 4.6 berikut.

Tabel 4.6 Temuan Kebutuhan Pihak Pelapor

No.	Temuan Kebutuhan
1	Pihak pelapor dapat melaporkan keluhan mengenai sarana prasarana di lingkup area FILKOM langsung kepada pegawai Bagian Perlengkapan tanpa harus mendatangi ruangan Bagian Perlengkapan.
2	Dalam melaporkan keluhan sarana prasarana, pihak pelapor dapat mengirimkan foto barang dan kode inventaris dari barang yang dilaporkan agar pelaporan dapat lebih spesifik dan lebih ringkas.
3	Pihak pelapor yang telah melapor dapat melacak status tindak-lanjut laporan dari keluhan yang telah disampaikan melalui notifikasi dalam aplikasi.

4	Pihak pelapor lain dapat melihat keluhan mengenai sarana prasarana apa saja yang telah dilaporkan beserta status tindak-lanjutnya.
5	Apabila setelah melaporkan keluhan sarana prasarana, laporan tersebut ditolak atau masih ditunda maka pihak pelapor juga dapat mengetahui alasan dari penolakan laporan keluhan tersebut.

Sedangkan dari hasil dan analisis wawancara yang dilakukan terhadap pihak *operator*, didapatkan temuan kebutuhan sistem untuk *operator* yang dijelaskan pada Tabel 4.7 berikut ini.

Tabel 4.7 Temuan Kebutuhan Pihak *Operator*

No.	Temuan Kebutuhan
1	Pelaporan keluhan sarana prasarana yang langsung diterima oleh pegawai Bagian Perlengkapan sebagai pihak <i>operator</i> .
2	Pihak <i>operator</i> dapat menerima laporan keluhan yang dikirim oleh pelapor, dapat memperbarui status tindak-lanjut laporan dan memberikan tanggapan dari laporan keluhan yang telah diterima.
3	Pembaharuan status tanggapan dari laporan keluhan yang telah diterima tidak hanya dapat dilihat oleh pelapor yang melaporkan, tetapi juga oleh seluruh civitas akademik lainnya.
4	Laporan keluhan sarana prasarana yang masuk dapat disertakan foto masalah yang terjadi dan kode inventaris apabila ada, untuk mempermudah dan mempercepat proses identifikasi.

4.1.3 Identifikasi Aktor

Tahap Identifikasi Aktor merupakan tahap untuk mengidentifikasi aktor-aktor yang akan berinteraksi dengan Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana FILKOM UB. Pada Tabel 4.8 berikut ini merupakan penjelasan dari aktor-aktor yang terlibat terhadap sistem beserta deskripsinya.

Tabel 4.8 Identifikasi Aktor

Aktor	Deskripsi
Pelapor	Pelapor merupakan aktor dari civitas akademik FILKOM yang dapat melaporkan keluhan sarana prasarana serta mendapat informasi status tanggapan dari keluhan yang telah dilaporkan.
<i>Operator</i>	<i>Operator</i> merupakan aktor dari pegawai Bagian Perlengkapan yang dapat menerima laporan keluhan sarana prasarana dari pelapor serta dapat memperbarui status laporan dan memberi tanggapan dari keluhan yang telah diterima.

4.1.4 Pemodelan Proses Bisnis *To-Be*

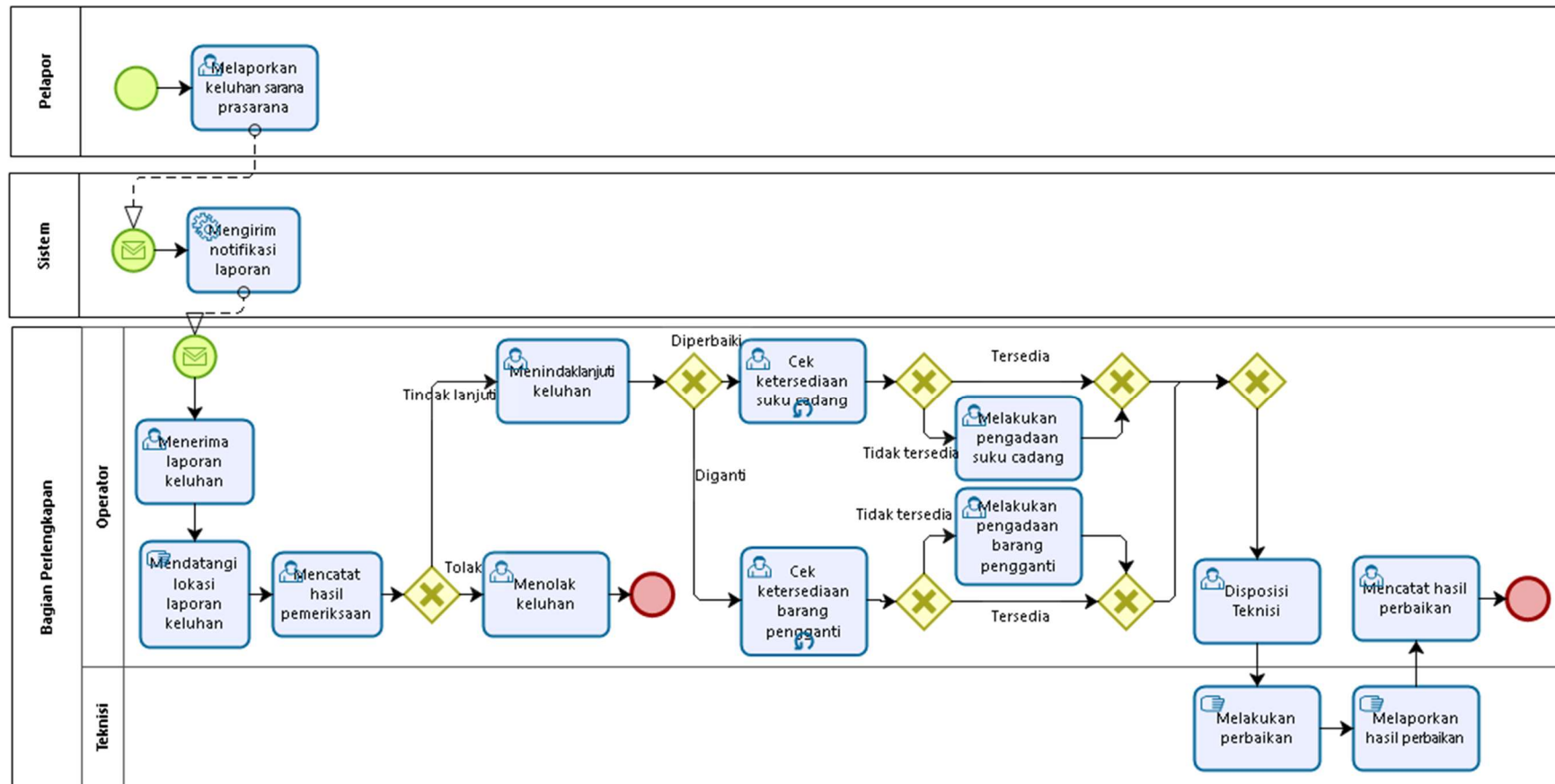
Pemodelan proses bisnis *to-be* merupakan penggambaran berupa model proses bisnis dari solusi yang diajukan untuk menyelesaikan permasalahan pada Tabel 4.5. Berdasarkan hasil analisis proses bisnis *as-is*, terdapat kesamaan aktivitas yang dilakukan oleh pegawai Bagian Perlengkapan ketika mereka telah menerima laporan keluhan mengenai sarana prasarana. Kesamaan itu antara lain adalah setelah menerima keluhan aktivitas selanjutnya adalah pemeriksaan sarana prasarana, pencatatan hasil pemeriksaan, cek ketersediaan suku cadang atau barang penggantinya, kemudian cek ketersediaan teknisi, dilakukan perbaikan dan setelah itu pencatatan kondisi terbaru. Sedangkan terdapat beberapa perbedaan yang dapat diketahui antara proses bisnis pelaporan keluhan secara langsung dan melalui E-Complaint. Perbedaan tersebut yang pertama adalah ketika melaporkan secara langsung baik secara lisan maupun melalui media komunikasi, pelapor tidak mendapatkan notifikasi mengenai status laporan dari keluhan yang sudah diterima. Perbedaan yang kedua adalah ketika melaporkan melalui E-Complaint, pelapor tidak dengan langsung berkomunikasi dengan pegawai Bagian Perlengkapan.

Berdasarkan hasil analisis permasalahan yang telah diperoleh, solusi terhadap permasalahan pelaporan keluhan mengenai sarana prasarana telah didapatkan dan dijelaskan dalam tabel daftar proses bisnis *to-be* sebagaimana pada Tabel 4.9 berikut. Daftar proses bisnis *to-be* akan dijelaskan berdasarkan nama proses bisnis disertai dengan kode dan deskripsinya dari masing-masing proses bisnis. Kode dari setiap proses bisnis akan diwakilkan dengan kode PB-SIMA-00. PB-SIMA merupakan singkatan dari Proses Bisnis Sistem Informasi Manajemen.

Tabel 4.9 Daftar Proses Bisnis *To-Be*

Kode Proses Bisnis	Nama	Deskripsi
PB-SIMA-01	Pelaporan Keluhan Sarana Prasarana	Proses bisnis dari pelaporan keluhan sarana prasarana yang disampaikan oleh pihak pelapor kepada <i>operator</i> melalui Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.
PB-SIMA-02	Pelacakan Status Tindak Lanjut Keluhan	Proses bisnis dari pelacakan status tindak lanjut keluhan yang telah dilaporkan oleh pihak pelapor kepada <i>operator</i> melalui Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.

Berdasarkan Tabel 4.9 yang berisikan daftar proses bisnis *to-be* beserta penjelasannya, telah diidentifikasi bahwa terdapat dua proses bisnis *to-be* yang perlu dimodelkan ke dalam diagram BPMN. Pemodelan dari proses bisnis PB-SIMA-01 ke dalam diagram BPMN akan ditunjukkan pada Gambar 4.3 berikut.

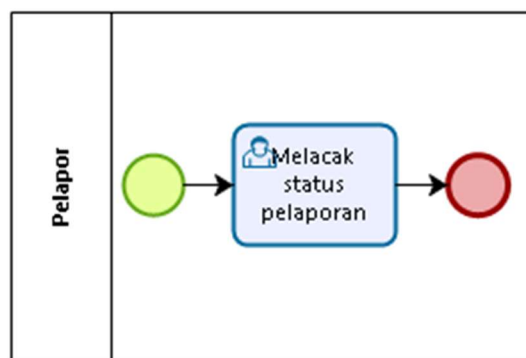


Gambar 4.3 Proses Bisnis To-Be PB-SIMA-01

Penjelasan dari Gambar 4.3 mengenai proses bisnis *to-be* PB-SIMA-01 antara lain adalah sebagai berikut:

1. Pelapor melaporkan keluhan mengenai sarana prasarana melalui aplikasi Sistem Informasi Manajemen Pelaporan.
2. Keluhan yang dilaporkan oleh pelapor diterima oleh sistem kemudian sistem mengirim notifikasi kepada operator mengenai adanya laporan keluhan baru.
3. Operator menerima notifikasi dan memeriksa berita keluhan yang telah dilaporkan.
4. Operator mendatangi lokasi dari laporan keluhan untuk mengadakan pemeriksaan masalah keluhan.
5. Operator memasukkan hasil dari pemeriksaan masalah keluhan ke dalam sistem kemudian operator menentukan untuk menindaklanjuti keluhan atau menolak keluhan.
6. Apabila operator menolak, maka operator mengubah status keluhan menjadi ditolak dan memasukkan alasan dari penolakan keluhan.
7. Apabila operator menindaklanjuti, maka operator mengubah status keluhan menjadi sedang ditindak lanjuti.
8. Operator menentukan apakah sarana prasarana tersebut memerlukan penggantian barang atau hanya penggantian suku cadang.
9. Operator memastikan ketersediaan suku cadang atau barang pengganti.
10. Apabila suku cadang atau barang pengganti tersedia maka dilanjutkan ke proses selanjutnya.
11. Apabila suku cadang atau barang pengganti tidak tersedia maka operator melakukan pengadaan suku cadang atau barang pengganti terlebih dahulu.
12. Operator melakukan disposisi teknisi kepada teknisi.
13. Teknisi melakukan perbaikan sarana prasarana.
14. Teknisi melaporkan hasil perbaikan sarana prasarana kepada operator.
15. Operator memasukkan hasil dari perbaikan yang telah dikerjakan oleh teknisi kemudian mengubah status keluhan menjadi selesai.

Selanjutnya pemodelan dari proses bisnis PB-SIMA-02 ke dalam diagram BPMN akan ditunjukkan pada Gambar 4.4 berikut.



Gambar 4.4 Proses Bisnis *To-Be* PB-SIMA-02

Penjelasan dari Gambar 4.4 mengenai proses bisnis *to-be* PB-SIMA-02 antara lain adalah sebagai berikut:

1. Pelapor melacak status tindak lanjut dari keluhan yang telah dilaporkan melalui Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana kemudian pelapor mendapatkan status tindak lanjut keluhan dari pilihan yang telah dipilih.

Berdasarkan proses bisnis PB-SIMA-01 yang ditunjukkan pada Gambar 4.3 dan proses bisnis PB-SIMA-02 yang ditunjukkan pada Gambar 4.4, selanjutnya akan disebutkan dan dijelaskan mengenai aktifitas-aktifitas yang akan didukung dengan penggunaan Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana ke dalam tabel daftar aktifitas proses bisnis. Tabel daftar aktifitas akan dijelaskan berdasarkan kode aktor kemudian kode aktifitas dan aktifitas yang dilakukan beserta dengan keterangan dari aktifitas tersebut. Daftar aktifitas proses bisnis PB-SIMA-01 akan dipaparkan dan dijelaskan sebagaimana pada Tabel 4.10 berikut.

Tabel 4.10 Daftar Aktifitas Proses Bisnis PB-SIMA-01

Kode Aktor	Kode Aktifitas	Aktifitas	Keterangan
Pelapor	SIMA-01-A01	Melaporkan keluhan sarana prasarana	Pelapor melaporkan keluhan mengenai sarana prasarana melalui Sistem Informasi Manajemen Aset
Sistem	SIMA-01-A02	Mengirim notifikasi laporan	Sistem mengirimkan notifikasi kepada operator mengenai adanya laporan keluhan baru.
Operator	SIMA-01-A03	Menerima laporan keluhan	Operator menerima notifikasi dari sistem kemudian memeriksa berita keluhan yang telah dilaporkan pelapor.

Kode Aktor	Kode Aktifitas	Aktifitas	Keterangan
	SIMA-01-A04	Mendatangi lokasi laporan keluhan	Operator mendatangi lokasi laporan keluhan untuk melakukan pemeriksaan.
	SIMA-01-A05	Mencatat hasil pemeriksaan	Operator memasukkan hasil dari pemeriksaan masalah keluhan ke dalam sistem.
	SIMA-01-A06	Menolak keluhan	Operator dapat memilih untuk menolak keluhan dan memberikan alasan penolakan.
	SIMA-01-A07	Menindaklanjuti keluhan	Operator dapat memilih untuk menindaklanjuti keluhan.
	SIMA-01-A08	Cek ketersediaan suku cadang	Operator memastikan ketersediaan suku cadang.
	SIMA-01-A09	Melakukan pengadaan suku cadang	Operator melakukan pengadaan suku cadang.
	SIMA-01-A10	Cek ketersediaan barang pengganti	Operator memastikan ketersediaan barang pengganti.
	SIMA-01-A11	Melakukan pengadaan barang pengganti	Operator melakukan pengadaan barang pengganti.
	SIMA-01-A12	Disposisi Teknisi	Operator mendisposisi teknisi untuk melakukan perbaikan.
	SIMA-01-A13	Melakukan Perbaikan	Teknisi melakukan perbaikan terhadap laporan keluhan.
	SIMA-01-A14	Mencatat hasil perbaikan	Operator memasukkan hasil dari perbaikan yang telah dikerjakan oleh teknisi kemudian mengubah status keluhan menjadi selesai.

Sedangkan daftar aktifitas proses bisnis PB-SIMA-02 akan dipaparkan dan dijelaskan sebagaimana pada Tabel 4.11 berikut.

Tabel 4.11 Daftar Aktifitas Proses Bisnis PB-SIMA-02

Kode Aktor	Kode Aktifitas	Aktifitas	Keterangan
Pelapor	SIMA-02-A01	Melacak status tindak lanjut	Pelapor memilih untuk melacak riwayat status tindak lanjut keluhan yang telah dilaporkan oleh pelapor tersebut atau melacak lini masa status tindak lanjut keluhan yang telah dilaporkan oleh pelapor lain.

4.1.5 Kebutuhan Fungsional

Kebutuhan fungsional dari sistem yang akan dibangun akan dijelaskan berdasarkan aktor-aktor yang berinteraksi dengan sistem disertai dengan kode, nama fungsi, deskripsi dari masing-masing kebutuhan dan kode aktifitas. Kode dari setiap kebutuhan fungsional dengan aktor pelapor akan diwakilkan dengan kode KFP-00-01, kebutuhan fungsional dengan aktor operator akan diwakilkan dengan kode KFO-00-01. KFP dan KFO secara berurutan merupakan singkatan dari Kebutuhan Fungsional Pelapor dan Kebutuhan Fungsional *Operator*, angka 00 merupakan penomoran kebutuhan utama dan angka 01 merupakan kebutuhan spesifik yang menjelaskan kebutuhan utama. Sedangkan kode aktifitas merupakan perwakilan dari aktifitas yang didapatkan dari table daftar aktifitas yang dijadikan sebagai acuan terhadap kemunculan masing-masing kebutuhan fungsional.

4.1.5.1 Kebutuhan Fungsional Pelapor

Penjabaran mengenai kebutuhan fungsional sistem dengan aktor pelapor akan ditunjukkan pada Tabel 4.12 berikut ini.

Tabel 4.12 Kebutuhan Fungsional Pelapor

Kode	Nama Fungsi	Deskripsi	Kode Aktifitas
KFP-01	Daftar Akun	<p>Sistem menyediakan halaman yang digunakan civitas FILKOM untuk mendaftarkan akun.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menampilkan <i>form</i> untuk daftar akun (<i>sign up</i>) dengan beberapa masukan. Masukan yang dibutuhkan adalah NIM/NIK/NIP, nama lengkap, <i>password</i>, nomor telepon, jenis pelapor dan jenis kelamin. (KFP-01-01) 	-

Kode	Nama Fungsi	Deskripsi	Kode Aktivitas
		<ul style="list-style-type: none"> Sistem akan meminta masukan berupa foto kartu penanda identitas untuk keperluan validasi pelapor. (KFP-01-02) Sistem tidak dapat melakukan pendaftaran akun pelapor apabila data pelapor yang diisikan belum lengkap. (KFP-01-03) 	
KFP-02	<i>Login</i>	<p>Sistem menyediakan halaman yang digunakan pelapor yang telah terdaftar untuk login.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Login dilakukan oleh pelapor dengan menggunakan NIM/NIK/NIP dan <i>password</i>. (KFP-02-01) Sistem tidak dapat login dan menampilkan peringatan kepada pelapor apabila NIM/NIK/NIP dan <i>password</i> masih ada yang kosong. (KFP-02-02) Sistem tidak dapat login dan menampilkan pesan kesalahan apabila NIM/NIK/NIP dan <i>password</i> tidak sesuai atau tidak terdaftar. (KFP-02-03) 	-
KFP-03	Lini masa laporan keluhan	<p>Sistem menyediakan halaman yang menampilkan keluhan-keluhan mengenai sarana prasarana yang telah dilaporkan oleh seluruh pihak pelapor yang terdaftar dan sedang ditindak lanjuti oleh pegawai Bagian Perlengkapan.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Lini Masa Laporan Keluhan pertama ditampilkan setelah login berhasil dilakukan. (KFP-03-01) Lini Masa Laporan Keluhan juga dapat dibuka melalui menu yang disediakan. (KFP-03-02) Lini Masa Laporan Keluhan menampilkan informasi mengenai laporan keluhan dari seluruh pihak pelapor terdaftar berupa judul keluhan, sarana prasarana yang bermasalah, deskripsi masalah, waktu dilaporkan, status 	SIMA-02-A01

Kode	Nama Fungsi	Deskripsi	Kode Aktifitas
		laporan, waktu dari tindak lanjutnya dan tanggapan dari pegawai Perlengkapan. (KFP-03-03)	
KFP-04	Kirim laporan keluhan	<p>Sistem menyediakan halaman yang digunakan pelapor untuk melaporkan dan mengirimkan keluhan kepada pegawai Bagian Perlengkapan.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> • Sistem menampilkan <i>form</i> untuk keluhan dengan beberapa masukan. Masukan yang dibutuhkan dari pelapor berupa judul keluhan, sarana prasarana yang bermasalah, deskripsi masalah dan saran perbaikan. (KFP-04-01) • Sistem menyediakan fitur pilihan agar pelapor dapat mengirimkan foto sarana prasarana yang bermasalah. (KFP-04-02) • Sistem menyediakan fitur pilihan agar pelapor dapat memindai kode inventaris sarana prasarana yang bermasalah menggunakan QR Code. (KFP-04-03) 	SIMA-01-A01
KFP-05	Tracking laporan keluhan	<p>Sistem menyediakan halaman untuk menampilkan dan melacak status keluhan yang telah dilaporkan oleh pelapor dan sedang ditindak lanjuti oleh pegawai Bagian Perlengkapan.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> • Sistem menampilkan informasi berupa ID keluhan, judul keluhan, sarana prasarana yang bermasalah, deskripsi masalah, saran perbaikan, waktu dilaporkan, status laporannya, waktu dari tindak lanjutnya dan tanggapan dari pegawai Perlengkapan. (KFP-05-01) 	SIMA-02-A01
KFP-06	Riwayat Laporan Keluhan	<p>Sistem menyediakan halaman untuk menampilkan keluhan yang pernah dilaporkan oleh pelapor dan telah selesai ditindak lanjuti pegawai Bagian Perlengkapan.</p> <p>Spesifikasi Kebutuhan:</p>	SIMA-02-A01

Kode	Nama Fungsi	Deskripsi	Kode Aktivitas
		<ul style="list-style-type: none"> Sistem menampilkan informasi berupa ID keluhan, judul keluhan, sarana prasarana yang bermasalah, deskripsi masalah, saran perbaikan, waktu dilaporkan, status laporannya, waktu dari tindak lanjutnya dan tanggapan dari pegawai Perlengkapan. (KFP-06-01) 	
KFP-07	Logout	<p>Sistem menyediakan fungsi yang digunakan pelapor yang telah <i>login</i> untuk melakukan <i>logout</i> atau keluar dari aplikasi.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menampilkan tombol <i>logout</i> di menu setting. (KFP-07-01) Sistem menampilkan halaman login setelah pelapor memilih <i>logout</i>. (KFP-07-02) 	-

4.1.5.2 Kebutuhan Fungsional *Operator*

Penjabaran mengenai kebutuhan fungsional sistem dengan aktor *Operator* akan ditunjukkan pada Tabel 4.13 berikut ini.

Tabel 4.13 Kebutuhan Fungsional *Operator*

Kode	Nama Fungsi	Deskripsi	Kode Aktivitas
KFO-01	Daftar Akun	<p>Sistem menyediakan halaman yang digunakan <i>operator</i> untuk mendaftarkan akun.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menampilkan <i>form</i> untuk mendaftarkan akun (<i>sign up</i>) dengan beberapa masukan. Masukan yang dibutuhkan adalah NIK/NIP, nama lengkap, jenis kelamin, jabatan, email, nomor telepon dan <i>password</i>. (KFO-01-01) Sistem tidak dapat melakukan pendaftaran akun <i>operator</i> apabila data <i>operator</i> yang diisikan belum lengkap. (KFO-01-02) 	-
KFO-02	Login	Sistem menyediakan halaman yang digunakan <i>operator</i> yang telah terdaftar untuk login.	-

Kode	Nama Fungsi	Deskripsi	Kode Aktivitas
		Spesifikasi Kebutuhan: <ul style="list-style-type: none"> • Login dilakukan oleh <i>operator</i> dengan menggunakan NIK/NIP dan <i>password</i>. (KFO-02-01) • Sistem tidak dapat login dan menampilkan peringatan kepada <i>operator</i> apabila NIK/NIP atau <i>password</i> masih ada yang kosong. (KFO-02-02) • Sistem tidak dapat login dan menampilkan pesan kesalahan kepada <i>operator</i> apabila NIK/NIP atau <i>password</i> tidak sesuai atau tidak terdaftar. (KFO-02-03) 	
KFO-03	Terima Laporan Keluhan	<p>Sistem menyediakan fitur bagi <i>operator</i> untuk menerima keluhan yang dilaporkan oleh pelapor.</p> Spesifikasi Kebutuhan: <ul style="list-style-type: none"> • Sistem menyediakan fitur berupa notifikasi yang diterima oleh <i>operator</i> ketika pelapor telah berhasil melaporkan keluhan. (KFO-03-01) • Sistem akan menampilkan rincian lengkap dari laporan keluhan ketika <i>operator</i> memilih notifikasi yang muncul. (KFO-03-02) 	SIMA-01-A03
KFO-04	Tindak Lanjut Keluhan	<p>Sistem menyediakan halaman bagi <i>operator</i> untuk melihat rincian lengkap dan menindak lanjuti laporan keluhan yang masuk.</p> Spesifikasi Kebutuhan: <ul style="list-style-type: none"> • Sistem menyediakan halaman yang menampilkan seluruh laporan keluhan sarana prasarana yang masuk, informasi yang ditampilkan antara lain ID keluhan, judul keluhan, NIM/NIK/NIP pelapor, ID aset, dan waktu dilaporkan. (KFO-04-01) • Sistem menyediakan halaman yang menampilkan informasi rincian lengkap dari laporan keluhan tertentu, informasi yang ditampilkan antara lain ID keluhan, judul keluhan, NIM/NIK/NIP pelapor, ID Aset, waktu 	SIMA-01-A05 sampai dengan SIMA-01-A14

Kode	Nama Fungsi	Deskripsi	Kode Aktivitas
		<p>dilaporkan, deskripsi masalah, usulan perbaikan dan foto keluhan. (KFO-04-02)</p> <ul style="list-style-type: none"> Sistem menyediakan fitur untuk memperbarui status laporan dari keluhan yang telah diterima. Status laporan yang tersedia antara lain diterima, sedang disurvei, menunggu suku cadang, menunggu barang pengganti, menunggu teknisi, sedang diperbaiki, ditunda, ditolak dan selesai. (KFO-04-03) Ketika status laporan adalah sedang disurvei, <i>operator</i> dapat memasukkan hasil pemeriksaan survey. (KFO-04-04) Ketika status laporan adalah selesai, <i>operator</i> dapat memasukkan hasil perbaikan. (KFO-04-05) Ketika status laporan adalah ditolak, <i>operator</i> dapat memasukkan alasan penolakan. (KFO-04-06) Sistem dapat meneruskan informasi perubahan status laporan dari laporan keluhan kepada pelapor dan juga dapat dilihat oleh seluruh civitas akademik FILKOM. (KFO-04-07) 	
KFO-05	Kelola Barang	<p>Sistem menyediakan halaman bagi <i>operator</i> untuk memeriksa ketersediaan, menambahkan, menghapus serta merubah jumlah ketersediaan suku cadang dan barang pengganti.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menyediakan halaman yang menampilkan daftar suku cadang dan barang pengganti beserta jumlahnya. (KFO-05-01) Sistem menyediakan halaman yang menampilkan informasi rincian lengkap dari suku cadang atau barang pengganti tertentu, informasi yang ditampilkan antara lain ID barang, nama barang, jenis barang dan jumlah barang. (KFO-05-02) 	<p>SIMA-01-A08</p> <p>sampai dengan</p> <p>SIMA-01-A11</p>

Kode	Nama Fungsi	Deskripsi	Kode Aktifitas
		<ul style="list-style-type: none"> Sistem menyediakan halaman untuk menambahkan suku cadang dan barang pengganti yang membutuhkan beberapa masukan yaitu nama barang dan jenis barang. (KFO-05-03) Sistem menyediakan halaman untuk merubah jumlah ketersediaan suku cadang dan barang pengganti. (KFO-05-04) Sistem menyediakan halaman untuk menghapus suku cadang dan barang pengganti. (KFO-05-05) 	
KFO-06	Kelola Teknisi	<p>Sistem menyediakan halaman bagi operator untuk menambahkan teknisi, menghapus teknisi dan melakukan disposisi teknisi untuk melakukan perbaikan sarana prasarana.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menyediakan halaman yang menampilkan daftar teknisi yang tersedia, informasi yang ditampilkan adalah ID teknisi, nama teknisi dan status ketersediaannya. (KFO-06-01) Sistem menyediakan halaman yang digunakan untuk mendaftarkan teknisi yang membutuhkan beberapa masukan yaitu ID teknisi, nama teknisi, jenis kelamin, alamat email, nomor telepon dan tempat kerja. (KFO-06-02) Sistem menyediakan halaman yang menampilkan informasi rincian lengkap dari teknisi tertentu, informasi yang ditampilkan antara lain adalah ID teknisi, nama teknisi, jenis kelamin, alamat email, nomor telepon, tempat kerja dan status ketersediaan. (KFO-06-03) Status ketersediaan menampilkan status pekerjaan yaitu antara lain tersedia, sedang bekerja dan sedang keluar. (KFO-06-04) 	SIMA-01-A12

Kode	Nama Fungsi	Deskripsi	Kode Aktivitas
		<ul style="list-style-type: none"> Sistem menyediakan fitur untuk melakukan disposisi teknisi untuk mengerjakan perbaikan. Teknisi yang dapat didisposisi hanya teknisi yang statusnya tersedia. Perbaikan yang dapat dikerjakan teknisi merupakan laporan yang telah disurvei oleh pegawai perlengkapan/<i>operator</i>. (KFO-06-05) Setelah perbaikan selesai dilakukan teknisi, <i>operator</i> dapat merubah status laporan menjadi selesai, memasukkan hasil perbaikan dan status ketersediaan teknisi kembali menjadi tersedia. (KFO-06-06) 	
KFO-07	Cek Daftar Pelapor	<p>Sistem menyediakan halaman bagi operator untuk melihat daftar pelapor.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menyediakan halaman yang menampilkan daftar pelapor. (KFO-07-01) Sistem menyediakan halaman yang menampilkan informasi rincian lengkap dari pelapor tertentu, informasi yang ditampilkan antara lain NIM/NIK/NIP, nama lengkap, nomor telepon, jenis pelapor dan jenis kelamin. (KFO-07-02) 	-
KFO-08	Logout	<p>Sistem menyediakan fungsi yang digunakan <i>operator</i> untuk melakukan <i>logout</i>.</p> <p>Spesifikasi Kebutuhan:</p> <ul style="list-style-type: none"> Sistem menampilkan tombol <i>logout</i> di menu setting. (KFO-08-01) Sistem menampilkan halaman login setelah pelapor memilih <i>logout</i>. (KFO-08-02) 	-

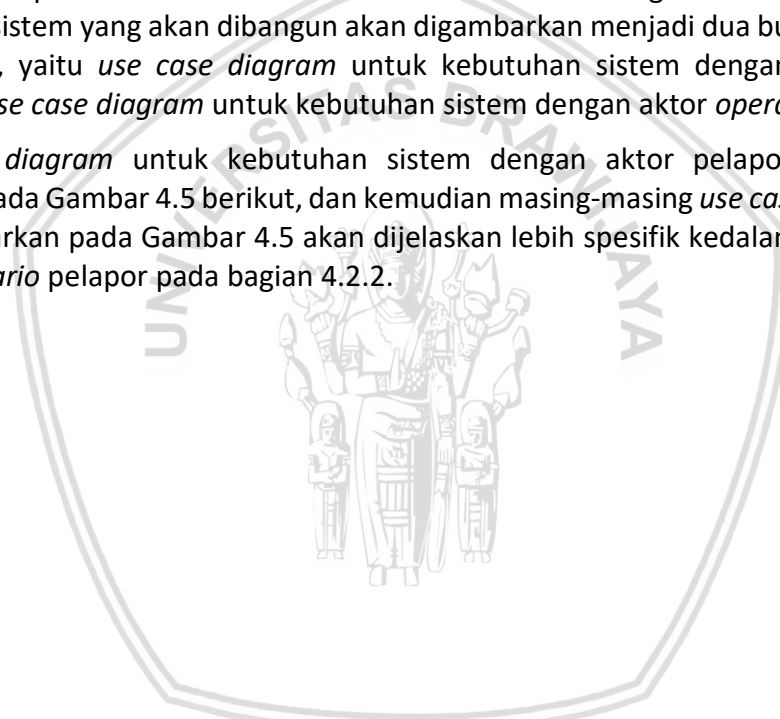
4.2 Pemodelan Kebutuhan

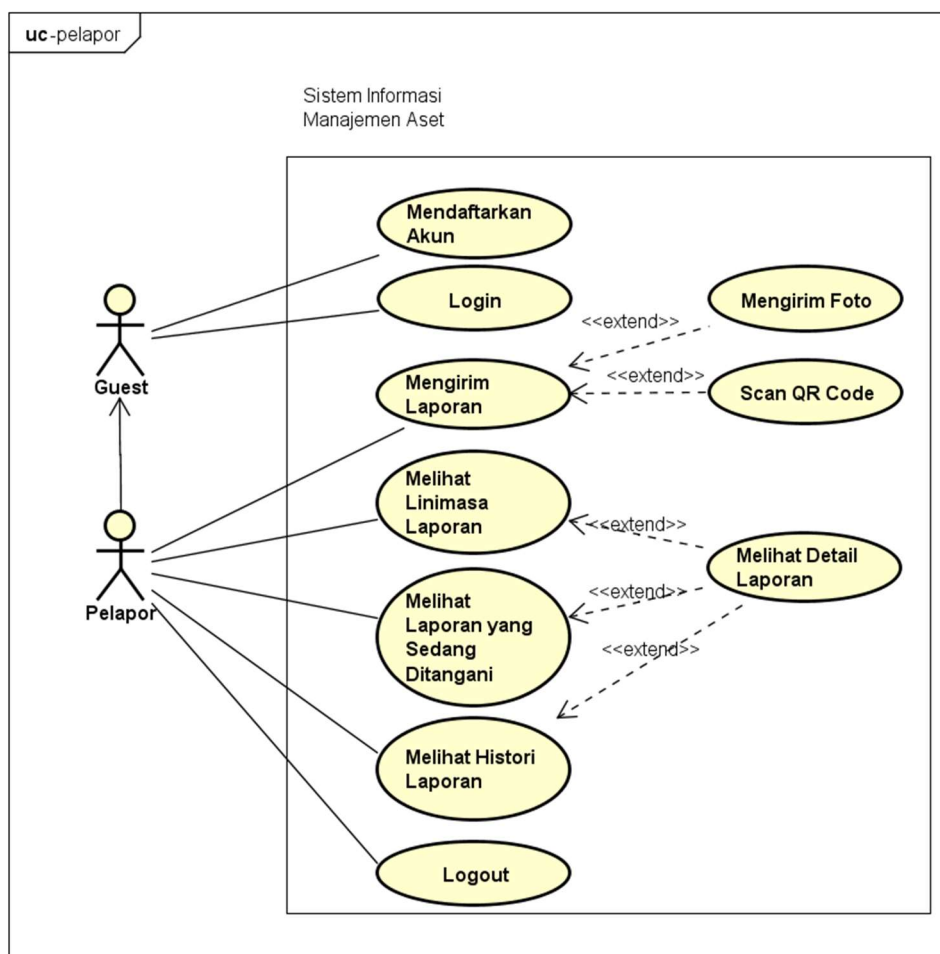
Tahap pemodelan kebutuhan merupakan tahap untuk memodelkan seluruh kebutuhan yang didapatkan pada tahap rekayasa kebutuhan ke dalam pemodelan Unified Modelling Language (UML). Dalam memodelkan kebutuhan ke dalam UML, pemodelan yang akan digunakan adalah *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

4.2.1 Pemodelan Use Case Diagram

Use case diagram merupakan diagram dari sudut pandang pengguna yang digunakan untuk mendeskripsikan fungsi-fungsi perilaku dan interaksi antara *stakeholder*, baik berupa pengguna atau sistem. Penggambaran *use case diagram* untuk memodelkan kebutuhan dari perilaku sistem akan mengacu kepada hasil dari pemodelan proses bisnis *to-be* dan analisis kebutuhan fungsional. *Use case diagram* dari sistem yang akan dibangun akan digambarkan menjadi dua buah *use case diagram*, yaitu *use case diagram* untuk kebutuhan sistem dengan aktor pelapor dan *use case diagram* untuk kebutuhan sistem dengan aktor *operator*.

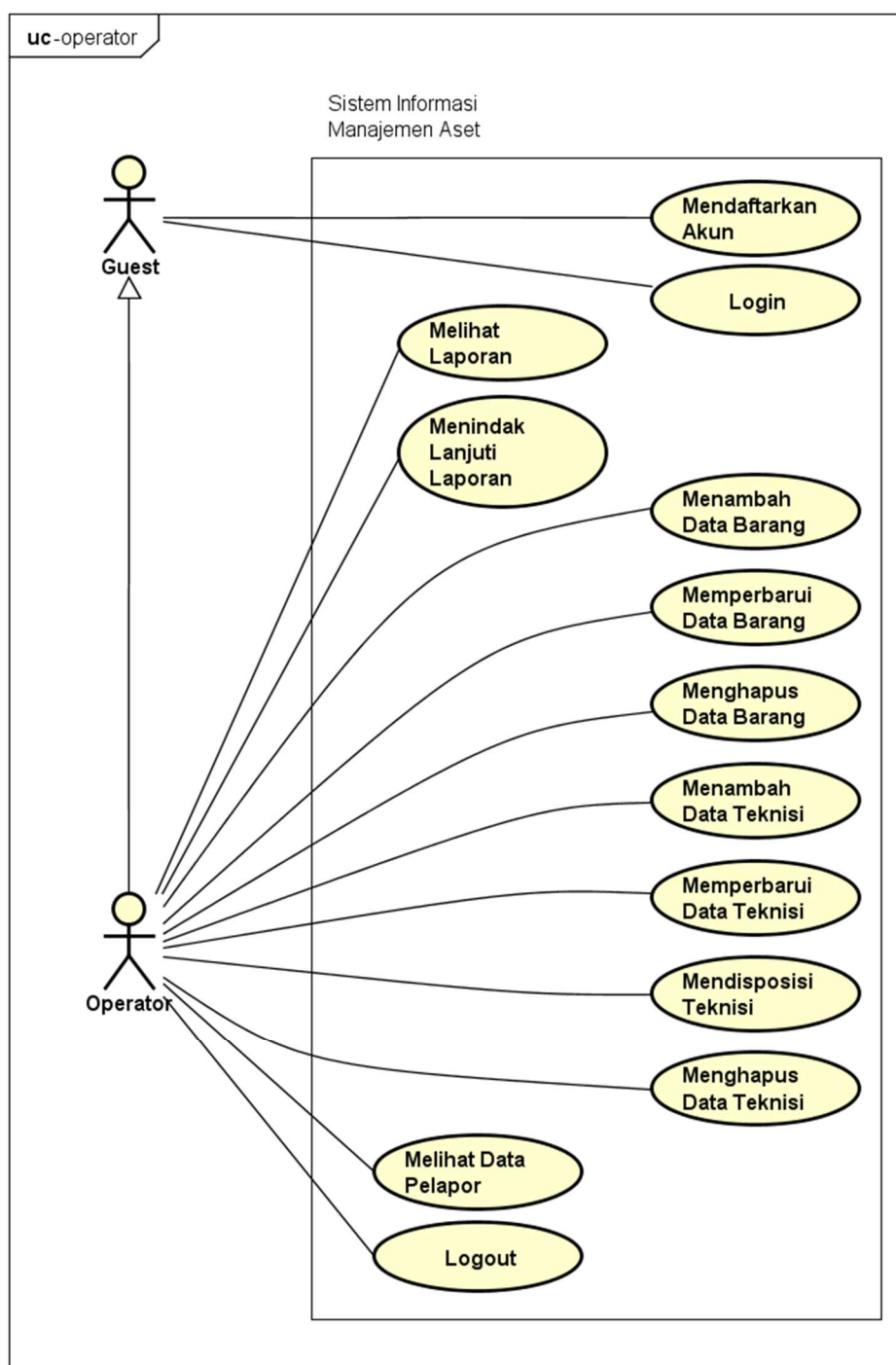
Use case diagram untuk kebutuhan sistem dengan aktor pelapor akan ditunjukkan pada Gambar 4.5 berikut, dan kemudian masing-masing *use case* yang telah digambarkan pada Gambar 4.5 akan dijelaskan lebih spesifik kedalam tabel *use case scenario* pelapor pada bagian 4.2.2.





Gambar 4.5 Use Case Diagram Pelapor Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana

Sedangkan *use case diagram* untuk kebutuhan sistem dengan aktor *operator* akan ditunjukkan pada Gambar 4.6 berikut, dan kemudian masing-masing *use case* yang telah digambarkan pada Gambar 4.6 akan dijelaskan lebih spesifik kedalam tabel *use case scenario operator* pada bagian 4.2.3.



Gambar 4.6 Use Case Diagram Operator Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana

4.2.2 Pemodelan *Use Case Scenario* Pelapor

4.2.2.1 Mendaftarkan Akun Pelapor

Tabel 4.14 Use Case Scenario Mendaftarkan Akun Pelapor

Kode Use Case	UCP-01
Nama	Mendaftarkan Akun
Aktor	<i>Guest</i>
Deskripsi	Sistem dapat menyimpan data akun pelapor agar <i>guest</i> mendapatkan hak akses terhadap sistem.
Kondisi Awal	Halaman <i>login</i> yang terdapat fungsi untuk membuka halaman mendaftar akun telah ditampilkan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Guest</i> memilih fungsi yang disediakan untuk membuka halaman mendaftar akun.	Sistem menampilkan halaman mendaftar akun yang berisi <i>form</i> untuk <i>sign up</i> .
2. <i>Guest</i> mengisi semua data masukan yang diperlukan untuk mendaftar akun pelapor baru kemudian memilih fungsi daftar akun baru.	Sistem menyimpan data akun pelapor kemudian mengarahkan <i>Guest</i> kembali ke halaman <i>Login</i> .
Skenario Alternatif	
3. <i>Guest</i> belum mengisikan semua data <i>form</i> yang harus dimasukkan.	Sistem memberikan pesan peringatan bahwa <i>form</i> masukan belum diisikan semua.
4. <i>Guest</i> sudah pernah mendaftar akun sebelumnya.	Sistem menampilkan pesan peringatan bahwa NIM pelapor telah terdaftar.
Kondisi Akhir	Akun pelapor berhasil diberikan kepada <i>guest</i> dan kembali diarahkan ke halaman <i>login</i>

4.2.2.2 Login Pelapor

Tabel 4.15 Use Case Scenario Login Pelapor

Kode Use Case	UCP-02
Nama	Login
Aktor	Guest
Deskripsi	Sistem dapat mencocokkan data NIM dan <i>password</i> akun pelapor ke dalam basis data agar <i>guest</i> bisa masuk ke dalam menu aplikasi Sistem Informasi Manajemen Aset.
Kondisi Awal	Halaman <i>login</i> yang terdapat <i>form</i> untuk memasukkan NIM/NIP/NIK dan <i>password</i> telah ditampilkan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Guest</i> memasukkan data NIM/NIP/NIK dan <i>password</i> ke dalam <i>form login</i> yang disediakan	Sistem memberikan akses ke dalam menu aplikasi Sistem Informasi Manajemen Aset
Skenario Alternatif	
2. <i>Guest</i> belum mengisi semua data <i>form login</i> yang harus dimasukkan.	Sistem memberikan pesan peringatan bahwa <i>form login</i> belum diisi semua.
3. <i>Guest</i> memasukkan data NIM atau <i>password</i> yang tidak sesuai.	Sistem memberikan pesan peringatan bahwa NIM atau <i>password</i> yang dimasukkan salah.
Kondisi Akhir	Menu aplikasi Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana ditampilkan kepada pelapor.

4.2.2.3 Mengirim Laporan

Tabel 4.16 Use Case Scenario Mengirim Laporan

Kode Use Case	UCP-03
Nama	Mengirim Laporan
Aktor	Pelapor
Deskripsi	Sistem dapat menyediakan <i>form</i> yang dapat diisi oleh pelapor untuk

	melaporkan keluhan sarana prasarana dan mengirimkannya ke <i>operator</i> .
Kondisi Awal	Menu utama aplikasi ditampilkan dan fungsi tampilkan halaman laporan baru disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. Pelapor mengakses halaman laporan baru.	Sistem menampilkan <i>form</i> untuk membuat laporan baru.
2. Pelapor mengisikan <i>form</i> untuk membuat laporan baru.	Sistem menerima masukan data judul keluhan, kode barang, deskripsi keluhan dan saran perbaikan.
3. Pelapor memilih fungsi <i>submit</i> untuk mengirimkan laporan.	Sistem mengirimkan masukan <i>form</i> laporan yang telah diisikan oleh pelapor ke <i>operator</i> .
Skenario Alternatif	
4. Pelapor dapat memilih fungsi gunakan kamera untuk mengambil foto keluhan.	Sistem menampilkan fitur kamera dari perangkat lalu menyimpan hasil foto keluhan.
5. Pelapor dapat memilih fungsi pindai <i>QR Code</i> untuk memindai kode barang apabila tersedia.	Sistem menampilkan fitur kamera pindai dari perangkat lalu menyimpan data kode barang hasil pembacaan <i>QR Code</i> .
6. Pelapor memilih fungsi <i>submit</i> ketika belum mengisikan data judul keluhan, kode barang, deskripsi keluhan dan saran perbaikan secara lengkap.	Sistem memberikan pesan peringatan bahwa <i>form</i> laporan baru belum diisikan semua.
Kondisi Akhir	Laporan keluhan baru dari pelapor berhasil dikirimkan kepada <i>operator</i> dan kembali diarahkan ke menu utama aplikasi.

4.2.2.4 Melihat Laporan Lini Masa

Tabel 4.17 Use Case Scenario Melihat Laporan Lini Masa

Kode Use Case	UCP-04
Nama	Melihat Laporan Lini Masa

Aktor	Pelapor
Deskripsi	Sistem dapat menyediakan halaman untuk menampilkan kumpulan keluhan-keluhan yang telah dilaporkan oleh pelapor lain.
Kondisi Awal	Menu utama aplikasi ditampilkan dan fungsi tampilkan daftar lini masa seluruh laporan disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. Pelapor memilih fungsi untuk menampilkan daftar lini masa seluruh keluhan.	Sistem menampilkan halaman lini masa seluruh keluhan yang telah dilaporkan oleh seluruh pelapor.
Skenario Alternatif	
2. Pelapor dapat memilih fungsi untuk melihat rincian lengkap dari masing-masing keluhan yang tersedia.	Sistem menampilkan halaman yang mengandung informasi rincian lengkap dari laporan keluhan yang terpilih.
Kondisi Akhir	Kumpulan keluhan-keluhan yang telah dilaporkan oleh pelapor lain berhasil didapatkan pelapor yang sedang <i>login</i> .

4.2.2.5 Melihat Laporan yang Sedang Ditangani

Tabel 4.18 Use Case Scenario Melihat Laporan yang Sedang Ditangani

Kode Use Case	UCP-05
Nama	Melihat Laporan yang Sedang Ditangani
Aktor	Pelapor
Deskripsi	Sistem dapat menyediakan halaman untuk menampilkan keluhan yang telah dilaporkan oleh Pelapor yang belum diselesaikan.
Kondisi Awal	Menu utama aplikasi ditampilkan dan fungsi tampilkan daftar laporan yang sedang ditangani disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem

1. Pelapor memilih fungsi untuk menampilkan daftar keluhan yang telah dilaporkan pelapor dan sedang ditangani.	Sistem menampilkan halaman daftar keluhan yang telah dilaporkan pelapor dan sedang ditangani.
Skenario Alternatif	
2. Pelapor dapat memilih fungsi untuk melihat rincian lengkap dari masing-masing keluhan yang tersedia.	Sistem menampilkan halaman yang mengandung informasi rincian lengkap dari laporan keluhan yang terpilih.
3. Pelapor sedang tidak ada keluhan yang dilaporkan dan belum diselesaikan atau ditolak	Sistem menampilkan pesan bahwa sedang tidak ada keluhan yang sedang dilaporkan.
Kondisi Akhir	Kumpulan keluhan yang telah dilaporkan oleh pelapor yang belum diselesaikan atau belum ditolak <i>operator</i> berhasil didapatkan.

4.2.2.6 Melihat Laporan Riwayat

Tabel 4.19 Use Case Scenario Melihat Laporan Riwayat

Kode Use Case	UCP-06
Nama	Melihat Laporan Riwayat
Aktor	Pelapor
Deskripsi	Sistem dapat menyediakan halaman untuk menampilkan keluhan yang pernah dilaporkan oleh Pelapor yang sudah diselesaikan atau sudah ditolak.
Kondisi Awal	Menu utama aplikasi ditampilkan dan fungsi tampilkan daftar laporan riwayat disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. Pelapor memilih fungsi untuk menampilkan daftar keluhan yang pernah dilaporkan oleh pelapor yang sudah diselesaikan atau sudah ditolak.	Sistem menampilkan halaman daftar keluhan yang pernah dilaporkan oleh pelapor yang sudah diselesaikan atau sudah ditolak.
Skenario Alternatif	

2. Pelapor dapat memilih fungsi untuk melihat rincian lengkap dari masing-masing keluhan yang tersedia.	Sistem menampilkan halaman yang mengandung informasi rincian lengkap dari laporan keluhan yang terpilih.
2. Pelapor tidak pernah melaporkan keluhan sebelumnya.	Sistem menampilkan pesan bahwa belum pernah ada keluhan yang dilaporkan.
Kondisi Akhir	Kumpulan keluhan yang telah dilaporkan oleh pelapor yang sudah diselesaikan atau sudah ditolak operator berhasil didapatkan.

4.2.2.7 Logout Pelapor

Tabel 4.20 Use Case Scenario Logout Pelapor

Kode Use Case	UCP-07
Nama	Logout
Aktor	Pelapor
Deskripsi	Sistem menyediakan tombol bagi Pelapor untuk keluar dan mengakhiri sesi hak akses Sistem Informasi Manajemen Aset.
Kondisi Awal	Fungsi <i>logout</i> disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. Pelapor memilih fungsi <i>logout</i> .	Sistem menampilkan pesan peringatan bahwa akan dilakukan <i>logout</i> .
2. Pelapor memilih pilihan <i>logout</i> .	Sistem mengakhiri hak akses Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana dari Pelapor dan kembali ke halaman <i>login</i> .
Skenario Alternatif	
3. Pelapor membatalkan pilihan <i>logout</i> .	Sistem membatalkan fungsi untuk <i>logout</i> dan kembali ke <i>menu</i> .
Kondisi Akhir	Sesi <i>login</i> terhadap aplikasi diakhiri sistem dan halaman <i>login</i> ditampilkan sistem.

4.2.3 Pemodelan Use Case Scenario Operator

4.2.3.1 Mendaftarkan Akun Operator

Tabel 4.21 Use Case Scenario Mendaftarkan Akun Operator

Kode Use Case	UCO-01
Nama	Mendaftarkan Akun
Aktor	Guest
Deskripsi	Sistem dapat menyimpan data akun <i>operator</i> agar <i>guest</i> mendapatkan hak akses terhadap sistem.
Kondisi Awal	Halaman <i>login</i> pada <i>website operator</i> dan fungsi untuk membuka halaman mendaftar akun ditampilkan oleh sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Guest</i> memilih fungsi yang disediakan untuk membuka halaman mendaftar akun.	Sistem menampilkan halaman mendaftar akun yang berisi <i>form</i> untuk <i>sign up</i> .
2. <i>Guest</i> mengisi semua data masukan yang diperlukan untuk mendaftar akun <i>operator</i> baru kemudian memilih fungsi daftar.	Sistem menyimpan data akun <i>operator</i> kemudian mengarahkan <i>guest</i> kembali ke halaman <i>Login</i> .
Skenario Alternatif	
3. <i>Guest</i> belum mengisikan semua data <i>form</i> yang harus dimasukkan.	Sistem memberikan pesan peringatan bahwa <i>form</i> masukan belum diisikan semua.
4. <i>Guest</i> sudah pernah mendaftar akun sebelumnya.	Sistem menampilkan pesan peringatan bahwa NIK <i>operator</i> telah terdaftar.
Kondisi Akhir	Akun <i>operator</i> berhasil diberikan kepada <i>guest</i> .

4.2.3.2 Login Operator

Tabel 4.22 Use Case Scenario Login Operator

Kode Use Case	UCO-02
Nama	Login

Aktor	Guest
Deskripsi	Sistem dapat mencocokkan data NIK dan <i>password</i> akun <i>operator</i> ke dalam basis data agar <i>guest</i> bisa masuk ke dalam <i>dashboard website operator</i> Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.
Kondisi Awal	Halaman <i>login</i> pada <i>website operator</i> yang terdapat <i>form</i> untuk memasukkan <i>username</i> dan <i>password</i> ditampilkan oleh sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Guest</i> membuka <i>website operator</i> .	Sistem menampilkan halaman <i>Login</i>
2. <i>Guest</i> memasukkan data NIK dan <i>password</i> ke dalam <i>form login</i> yang disediakan.	Sistem memberikan akses ke dalam <i>dashboard website operator</i> Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana.
Skenario Alternatif	
3. <i>Guest</i> belum mengisikan semua data <i>form login</i> yang harus dimasukkan.	Sistem memberikan pesan peringatan bahwa <i>form login</i> belum diisikan semua.
4. <i>Guest</i> memasukkan data NIK atau <i>password</i> yang tidak sesuai.	Sistem memberikan pesan peringatan bahwa NIK atau <i>password</i> yang dimasukkan salah.
Kondisi Akhir	Menu utama <i>website operator</i> ditampilkan kepada <i>operator</i> .

4.2.3.3 Melihat Laporan

Tabel 4.23 Use Case Scenario Melihat Laporan

Kode Use Case	UCO-03
Nama	Melihat Laporan
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fitur dan halaman bagi <i>operator</i> untuk melihat laporan masuk yang dilaporkan oleh pelapor.

Kondisi Awal	Menu utama <i>website operator</i> dan fungsi tampilkan seluruh laporan ditampilkan oleh sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memilih fungsi untuk membuka halaman seluruh laporan.	Sistem menampilkan daftar seluruh laporan yang masuk.
2. <i>Operator</i> memilih fungsi lihat rincian laporan tertentu.	Sistem menampilkan informasi rincian lengkap dari laporan yang terpilih dari seluruh laporan.
Skenario Alternatif	
3. <i>Operator</i> memilih fungsi lihat notifikasi masuk.	Sistem menampilkan tampilan notifikasi laporan yang masuk.
4. <i>Operator</i> memilih fungsi lihat rincian laporan tertentu.	Sistem menampilkan informasi rincian lengkap dari laporan yang terpilih melalui notifikasi.
Kondisi Akhir	Informasi rincian mengenai laporan yang telah dilaporkan oleh pelapor ditampilkan kepada operator.

4.2.3.4 Menindak Lanjuti Laporan

Tabel 4.24 Use Case Scenario Menindak Lanjuti Laporan

Kode Use Case	UCO-04
Nama	Menindak Lanjuti Laporan
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk mengubah status laporan dan memasukkan keterangan dari perubahan status tindak lanjut.
Kondisi Awal	Informasi rincian lengkap dari laporan yang telah dipilih ditampilkan kepada operator.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> merubah status tindak lanjut menjadi sedang disurvey.	Sistem menyimpan perubahan status dan menampilkan <i>form</i> masukan hasil survey.

2. <i>Operator</i> memasukkan hasil survey laporan ke dalam sistem.	Sistem menyimpan hasil survey laporan dari <i>operator</i> .
3. <i>Operator</i> merubah status tindak lanjut menjadi sedang diperbaiki.	Sistem menampilkan daftar teknisi yang tersedia.
4. <i>Operator</i> melakukan disposisi teknisi.	Sistem menyimpan perubahan status menjadi sedang diperbaiki dan menyimpan data teknisi yang diberikan disposisi.
5. <i>Operator</i> merubah status tindak lanjut menjadi selesai.	Sistem menyimpan perubahan status menjadi selesai, merubah status teknisi yang melakukan perbaikan menjadi tersedia dan menampilkan <i>form</i> masukan hasil akhir.
6. <i>Operator</i> memasukkan hasil akhir perbaikan.	Sistem menyimpan masukan hasil akhir perbaikan dari operator.
Skenario Alternatif	
7. <i>Operator</i> merubah status tindak lanjut menjadi menunggu suku cadang.	Sistem menyimpan perubahan status menjadi menunggu suku cadang.
8. <i>Operator</i> merubah status tindak lanjut menjadi menunggu barang pengganti.	Sistem menyimpan perubahan status menjadi menunggu barang pengganti.
9. <i>Operator</i> merubah status tindak lanjut menjadi menunggu teknisi	Sistem menyimpan perubahan status menjadi menunggu teknisi.
10. <i>Operator</i> merubah status tindak lanjut menjadi ditunda.	Sistem menyimpan perubahan status menjadi ditunda dan menampilkan <i>form</i> masukan alasan penundaan.
11. <i>Operator</i> memasukkan alasan penundaan keluhan.	Sistem menyimpan masukan alasan penundaan keluhan dari operator.
12. <i>Operator</i> merubah status tindak lanjut menjadi ditolak.	Sistem menyimpan perubahan status menjadi ditolak dan menampilkan <i>form</i> masukan alasan penolakan.
13. <i>Operator</i> memasukkan alasan penolakan keluhan.	Sistem menyimpan masukan alasan penolakan dari operator.

Kondisi Akhir	Perubahan data laporan yang telah ditindaklanjuti berhasil disimpan sistem dan informasi rincian lengkap dari laporan tersebut ditampilkan sistem.
----------------------	--

4.2.3.5 Menambah Data Barang

Tabel 4.25 Use Case Scenario Menambah Data Barang

Kode Use Case	UCO-05
Nama	Menambah Data Barang
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk menambah data barang sarana prasarana.
Kondisi Awal	Fungsi tambah barang disediakan sistem lalu halaman <i>form</i> tambah barang ditampilkan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memasukkan data barang yang akan ditambahkan berdasarkan <i>form</i> tambah barang yang tersedia.	Sistem menerima masukan data barang yang telah dimasukkan oleh operator.
2. <i>Operator</i> memilih fungsi yang disediakan untuk menambahkan data barang berdasarkan data yang telah dimasukkan.	Sistem menyimpan masukan data barang untuk menambahkan data barang baru.
Skenario Alternatif	
3. <i>Operator</i> belum mengisikan semua data <i>form</i> tambah barang yang harus dimasukkan.	Sistem memberikan pesan peringatan bahwa <i>form</i> tambah barang belum diisikan semua.
4. Data barang yang dimasukkan oleh <i>operator</i> sudah pernah dibuat sebelumnya.	Sistem memberikan pesan peringatan bahwa data barang yang dimasukkan sudah pernah dibuat sebelumnya.
Kondisi Akhir	Data barang sarana prasarana baru berhasil disimpan dan halaman

	seluruh data barang ditampilkan sistem.
--	---

4.2.3.6 Memperbarui Data Barang

Tabel 4.26 Use Case Scenario Memperbarui Data Barang

Kode Use Case	UCO-06
Nama	Memperbarui Data Barang
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk memperbarui data barang sarana prasarana.
Kondisi Awal	Daftar seluruh barang dan fungsi untuk memperbarui barang ditampilkan oleh sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memilih data barang yang akan diperbarui.	Sistem menampilkan data lengkap dari barang yang dipilih <i>operator</i> .
2. <i>Operator</i> merubah dan memperbarui data dari barang yang telah dipilih.	Sistem menerima masukan perubahan data dari operator kemudian menampilkan pesan peringatan mengenai pembaharuan data.
3. <i>Operator</i> memilih fungsi untuk menyimpan pembaharuan.	Sistem menyimpan pembaharuan data barang yang telah dipilih.
Skenario Alternatif	
4. <i>Operator</i> membatalkan fungsi pembaharuan data barang.	Sistem menerima pembatalan dari <i>operator</i> dan tidak menyimpan pembaharuan data barang yang dipilih
Kondisi Akhir	Pembaharuan data barang sarana prasarana berhasil disimpan sistem dan data lengkap dari barang tersebut ditampilkan.

4.2.3.7 Menghapus Data Barang

Tabel 4.27 Use Case Scenario Menghapus Data Barang

Kode Use Case	UCO-07
----------------------	--------

Nama	Menghapus Data Barang
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk menghapuskan data barang sarana prasarana.
Kondisi Awal	Daftar seluruh barang ditampilkan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memilih data barang yang akan dihapus.	Sistem menampilkan pesan peringatan kepada <i>operator</i> untuk penghapusan data barang yang dipilih.
2. <i>Operator</i> memilih fungsi hapus data barang.	Sistem menghapus data barang yang telah dipilih oleh <i>operator</i> .
Skenario Alternatif	
3. <i>Operator</i> membatalkan penghapusan data barang yang telah dipilih.	Sistem menerima pembatalan dari <i>operator</i> dan tidak menghapus data barang yang dipilih.
Kondisi Akhir	Data barang sarana prasarana berhasil dihapus dan halaman seluruh data barang ditampilkan sistem.

4.2.3.8 Menambah Data Teknisi

Tabel 4.28 Use Case Scenario Menambah Data Teknisi

Kode Use Case	UCO-08
Nama	Menambah Data Teknisi
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk menambahkan data teknisi.
Kondisi Awal	Fungsi tambah teknisi dan <i>form</i> tambah teknisi ditampilkan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memasukkan data teknisi yang akan ditambahkan	Sistem menerima masukan data teknisi yang telah dimasukkan oleh <i>operator</i> .

berdasarkan <i>form</i> tambah teknisi yang tersedia.	
2. <i>Operator</i> memilih fungsi yang disediakan untuk menambahkan data teknisi berdasarkan data yang telah dimasukkan.	Sistem menyimpan masukan data teknisi untuk menambahkan data teknisi baru.
Skenario Alternatif	
3. <i>Operator</i> belum mengisikan semua data <i>form</i> tambah teknisi yang harus dimasukkan.	Sistem memberikan pesan peringatan bahwa <i>form</i> tambah teknisi belum diisikan semua.
4. Data teknisi yang dimasukkan oleh <i>operator</i> sudah pernah dibuat sebelumnya.	Sistem memberikan pesan peringatan bahwa data teknisi yang dimasukkan sudah pernah dibuat sebelumnya.
Kondisi Akhir	Data teknisi baru berhasil disimpan oleh sistem.

4.2.3.9 Memperbarui Data Teknisi

Tabel 4.29 Use Case Scenario Memperbarui Data Teknisi

Kode Use Case	UCO-09
Nama	Memperbarui Data Teknisi
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk memperbarui data teknisi.
Kondisi Awal	Daftar seluruh teknisi dan fungsi untuk memperbarui data teknisi tertentu ditampilkan oleh sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memilih teknisi yang akan diperbarui datanya.	Sistem menampilkan informasi lengkap dari teknisi yang dipilih oleh <i>operator</i> .
2. <i>Operator</i> merubah dan memperbarui data dari teknisi yang telah dipilih.	Sistem menerima masukan perubahan data dari <i>operator</i> kemudian menampilkan pesan peringatan mengenai pembaharuan data.
Skenario Alternatif	

3. <i>Operator</i> membatalkan fungsi pembaharuan data teknisi.	Sistem menerima pembatalan dari <i>operator</i> dan tidak menyimpan pembaharuan data teknisi yang dipilih.
Kondisi Akhir	Pembaharuan data teknisi yang terpilih berhasil disimpan dan data lengkap dari teknisi tersebut ditampilkan sistem.

4.2.3.10 Mendisposisi Teknisi

Tabel 4.30 Use Case Scenario Mendisposisi Teknisi

Kode Use Case	UCO-10
Nama	Mendisposisi Teknisi
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk mendisposisi teknisi.
Kondisi Awal	Fungsi menampilkan halaman daftar seluruh teknisi disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> membuka halaman yang menampilkan seluruh teknisi.	Sistem menampilkan daftar seluruh teknisi beserta dengan status ketersediaannya.
2. <i>Operator</i> memilih teknisi yang akan dilakukan disposisi.	Sistem menampilkan daftar keluhan yang dapat dilakukan perbaikan oleh teknisi, yaitu keluhan yang telah disurvei dan disimpan hasil surveynya.
3. <i>Operator</i> memilih keluhan yang harus dilakukan perbaikan oleh teknisi.	Sistem menerima pilihan keluhan yang harus dilakukan perbaikan kemudian merubah status tindak lanjut keluhan tersebut menjadi sedang diperbaiki serta merubah status ketersediaan teknisi tersebut dari tersedia menjadi sedang bekerja.
Skenario Alternatif	
4. Teknisi yang dilakukan disposisi statusnya sedang bekerja atau sedang keluar.	Sistem tidak dapat menerima pilihan disposisi oleh <i>operator</i> .

Kondisi Akhir	Disposisi teknisi terhadap keluhan terpilih berhasil dilakukan kemudian daftar seluruh teknisi dan status ketersediannya ditampilkan sistem.
----------------------	--

4.2.3.11 Menghapus Data Teknisi

Tabel 4.31 Use Case Scenario Menghapus Data Teknisi

Kode Use Case	UCO-011
Nama	Menghapus Data Teknisi
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi untuk menghapuskan data teknisi.
Kondisi Awal	Daftar seluruh teknisi ditampilkan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memilih teknisi yang datanya akan dihapus.	Sistem menampilkan pesan peringatan kepada <i>operator</i> untuk penghapusan data teknisi yang dipilih.
2. <i>Operator</i> memilih fungsi hapus data teknisi.	Sistem menghapus data teknisi yang telah dipilih oleh <i>operator</i> .
Skenario Alternatif	
3. <i>Operator</i> membatalkan penghapusan data teknisi yang telah dipilih.	Sistem menerima pembatalan dari <i>operator</i> dan tidak menghapus data teknisi yang dipilih.
Kondisi Akhir	Data teknisi berhasil dihapus sistem kemudian halaman seluruh data teknisi ditampilkan.

4.2.3.12 Melihat Daftar Pelapor

Tabel 4.32 Use Case Scenario Melihat Daftar Pelapor

Kode Use Case	UCO-12
Nama	Melihat Daftar Pelapor
Aktor	<i>Operator</i>

Deskripsi	Sistem menyediakan halaman untuk menampilkan daftar pelapor dan informasi lengkapnya
Kondisi Awal	Fungsi menampilkan halaman daftar seluruh pelapor disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. Operator memilih fungsi untuk menampilkan halaman daftar seluruh pelapor.	Sistem menampilkan halaman yang berisi daftar seluruh pelapor.
Skenario Alternatif	
2. Operator memilih fungsi untuk menampilkan informasi lengkap dari pelapor yang dipilih.	Sistem menampilkan halaman yang berisi informasi lengkap mengenai pelapor yang dipilih.
Kondisi Akhir	Daftar pelapor dan informasi lengkapnya ditampilkan sistem.

4.2.3.13 Logout Operator

Tabel 4.33 Use Case Scenario Logout Operator

Kode Use Case	UCO-13
Nama	Logout
Aktor	<i>Operator</i>
Deskripsi	Sistem menyediakan fungsi bagi <i>operator</i> untuk keluar dan mengakhiri sesi hak akses <i>website operator</i> Sistem Informasi Manajemen Aset.
Kondisi Awal	Fungsi untuk <i>logout</i> disediakan sistem.
Skenario Utama	
Aksi Aktor	Respon Sistem
1. <i>Operator</i> memilih fungsi <i>logout</i>	Sistem menampilkan pesan peringatan bahwa operator akan melakukan <i>logout</i> .
2. <i>Operator</i> memilih pilihan <i>logout</i> .	Sistem mengakhiri hak akses <i>website operator</i> Sistem Informasi Manajemen Aset dari <i>operator</i> dan kembali ke halaman <i>login</i> .

Skenario Alternatif	
3. <i>Operator</i> membatalkan pilihan <i>logout</i> .	Sistem membatalkan fungsi untuk <i>logout</i> dan kembali ke <i>menu dashboard</i> .
Kondisi Akhir	Sesi <i>login</i> terhadap <i>website operator</i> diakhiri oleh sistem dan halaman <i>login</i> ditampilkan.

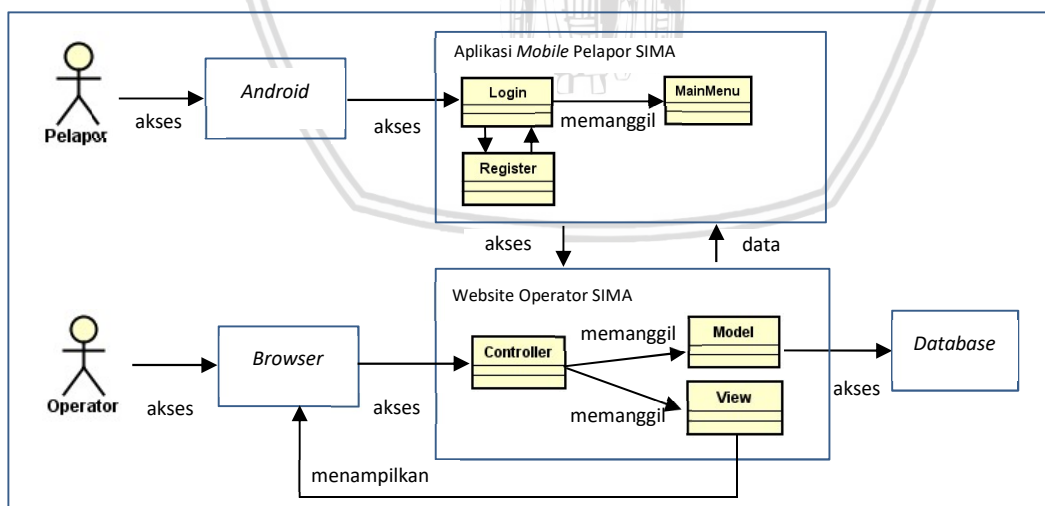


BAB 5 PERANCANGAN

Perancangan merupakan representasi dari Tahap Perancangan pada *Waterfall Model*, yang mana merupakan proses penentuan rencana dari solusi yang mencakup perancangan arsitektur perangkat lunak, perancangan *sequence diagram*, perancangan *class diagram*, perancangan skema basis data serta desain grafis antarmuka pengguna oleh pengembang perangkat lunak. Pada proses perancangan sistem dari perancangan *sequence diagram* kemudian *class diagram* hingga skema basis data, pemodelan yang akan digunakan adalah menggunakan bahasa pemodelan *Unified Modelling Language (UML)*. UML yang akan digunakan untuk proses ini terdiri dari *Sequence Diagram* dan *Class Diagram*. Kemudian dalam menggambarkan skema basis data akan dilakukan analisis dan pemetaan berdasarkan perancangan yang telah didapatkan pada *Class Diagram*.

5.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem merupakan tahap pertama yang harus dilakukan dalam proses perancangan perangkat lunak. Dalam proses pengembangan suatu sistem, perancangan arsitektur sistem digunakan untuk menjelaskan komponen struktur utama dari sistem dan hubungan antar komponen-komponen tersebut. Pada bagian ini arsitektur sistem bertujuan untuk menggambarkan bagaimana interaksi antar komponen-komponen Sistem Informasi Manajemen (SIMA) dalam membantu civitas akademik FILKOM sebagai pelapor untuk melaporkan keluhan mengenai sarana prasarana kepada pegawai Bagian Perlengkapan sebagai *operator*. Perancangan arsitektur Sistem Informasi Manajemen dapat dilihat pada Gambar 5.1 berikut.



Gambar 5.1 Architecture Diagram Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana

Pada Gambar 5.1 menjelaskan bahwa pelapor dapat melaporkan keluhan mengenai sarana prasarana dengan cara mengakses aplikasi *mobile* pelapor SIMA pada perangkat bersistem operasi *android*. Aplikasi *mobile* pelapor SIMA yang dibangun dikembangkan menggunakan bahasa pemrograman Java. Untuk keperluan mengakses data ke dalam dan dari basis data, Aplikasi *mobile* pelapor SIMA memerlukan perantara berupa *web service* menggunakan bahasa pemrograman sisi server untuk melakukan koneksi dan pengaksesan data yang dapat diakses dari *Website Operator* SIMA. Bahasa pemrograman sisi server yang digunakan untuk membangun *web service* adalah menggunakan bahasa PHP. Sedangkan untuk *operator* dalam menerima laporan keluhan dan menindaklanjuti laporan tersebut perlu mengakses *website operator* melalui *browser* terlebih dahulu. *Website operator* SIMA dikembangkan dengan menerapkan *framework Codeigniter* yang mana *framework* ini menerapkan pola perancangan *Model-View-Controller* (MVC).

5.2 Perancangan *Sequence Diagram*

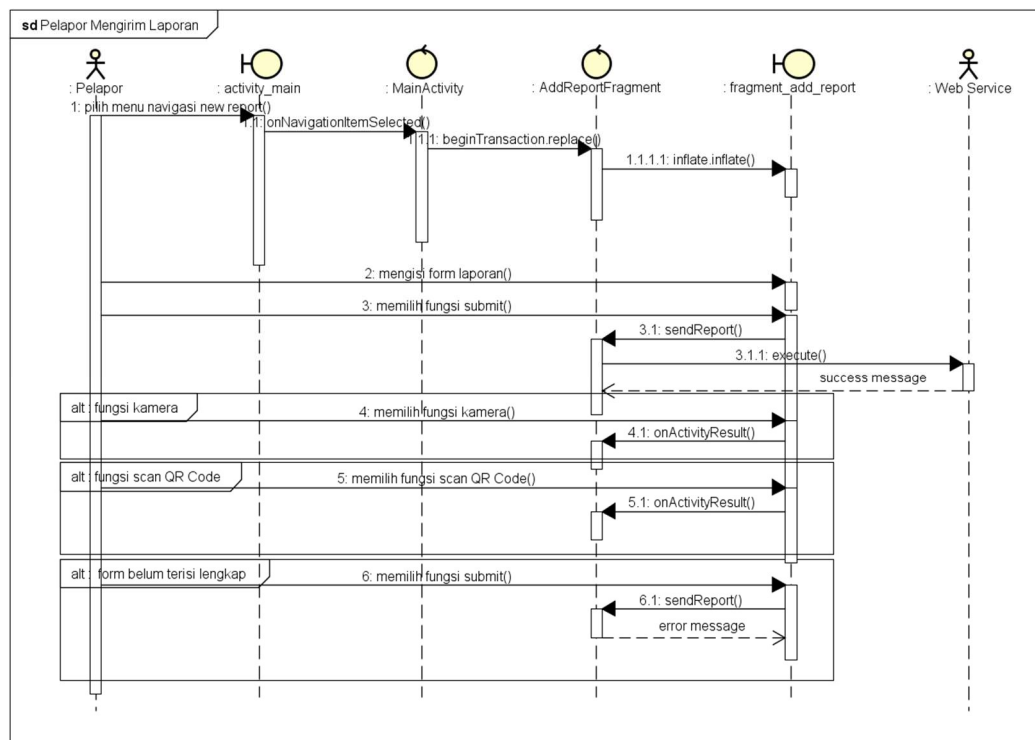
Pada bagian ini berisi penggambaran alur proses yang terjadi di dalam Sistem Informasi Manajemen Aset. Pada perancangan *sequence diagram* berikut ini akan dibagi menjadi dua bagian sesuai aktor yang telah diidentifikasi pada tahap analisis, yaitu perancangan *sequence diagram* pelapor dan perancangan *sequence diagram operator*.

5.2.1 Perancangan *Sequence Diagram* Pelapor

Pada bagian ini merupakan penggambaran *sequence diagram* dengan aktor pelapor. *Sequence diagram* yang digambarkan pada bagian ini mengacu kepada *use case diagram* uc-pelapor.

5.2.1.1 *Sequence Diagram* Mengirim Laporan

Sequence diagram mengirim laporan mengacu pada *use case* dengan kode UCP-03 dan dapat dilihat pada Gambar 5.2. Alur ini diawali dengan pelapor memilih fungsi laporan baru pada menu navigasi. Kemudian setelah itu tampilan *activity_main* akan menjalankan fungsi *onNavigationItemSelectedListener()* pada *MainActivity* untuk mengganti *fragment* yang sedang dimuat di *MainActivity* menjadi *AddReportFragment* lalu *fragment* tersebut memuat tampilan *fragment_add_report* yang berisi *form-form* untuk melakukan pelaporan keluhan. Setelah itu pelapor dapat mengisi data pada *form* yang tersedia, kemudian ketika pelapor memilih fungsi *submit*, *fragment_add_report* akan menjalankan fungsi *sendReport* pada *AddReportFragment* untuk mengeksekusi pengiriman data laporan kepada *Web Service*. Setelah data laporan berhasil dikirimkan *Web Service* mengirimkan pesan sukses kepada *AddReportFragment*.



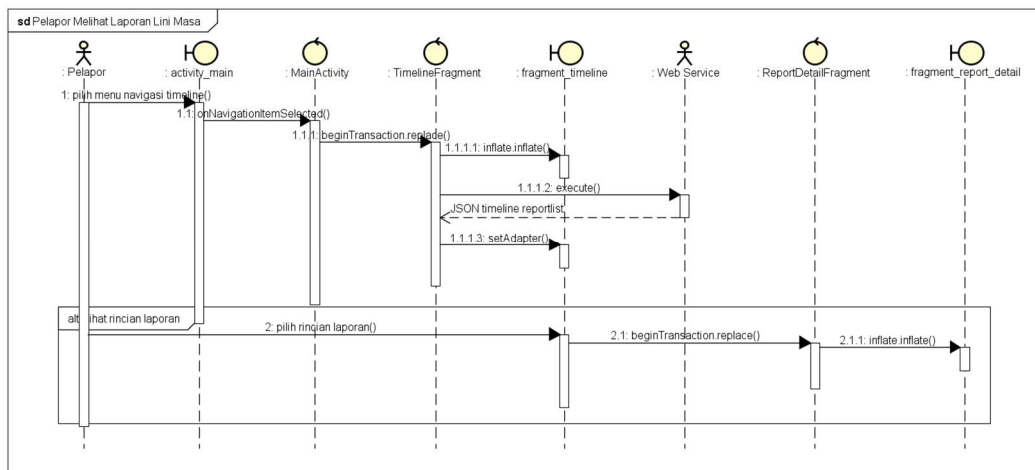
Gambar 5.2 Sequence Diagram Mengirim Laporan

Gambar 5.2 juga menunjukkan alur alternatif ketika pelapor memilih fungsi kamera pada saat mengisi *form* laporan keluhan untuk mengambil foto keluhan, maka *fragment_add_report* akan menjalankan fungsi *onActivityResult()* untuk menyimpan sementara hasil tangkapan foto tersebut. Alur alternatif selanjutnya adalah ketika pelapor memilih fungsi *scan QR Code* pada saat mengisi *form* laporan keluhan untuk mengisi kode aset, maka *fragment_add_report* akan menjalankan fungsi *onActivityResult()* untuk menyimpan data hasil terjemahan *QR Code* dan mengisi data tersebut ke dalam *form* kode aset. Alur alternatif yang terakhir adalah ketika pelapor memilih fungsi *submit* laporan sedangkan data pada *form* laporan belum terisi semua, maka ketika fungsi *sendReport* pada *fragment_add_report* dijalankan akan mengembalikan pesan *error* kepada pelapor bahwa *form* belum terisi semua.

5.2.1.2 Sequence Diagram Melihat Laporan Lini Masa

Sequence diagram melihat laporan lini masa mengacu pada *use case* dengan kode UCP-04 dan dapat dilihat pada Gambar 5.3. Alur ini diawali dengan pelapor memilih fungsi lini masa pada menu navigasi. Kemudian setelah itu tampilan *activity_main* akan menjalankan fungsi *onNavigationItemSelected()* pada *MainActivity* untuk mengganti *fragment* yang sedang dimuat di *MainActivity* menjadi *TimelineFragment* lalu *fragment* tersebut memuat tampilan *fragment_timeline* yang berisi daftar seluruh laporan keluhan yang telah dilaporkan oleh pelapor-pelapor yang telah terdaftar. Selanjutnya *TimelineFragment* menjalankan fungsi *execute()* untuk meminta data seluruh laporan keluhan kepada *Web Service* yang berperan sebagai perantara dengan

basis data SIMA, kemudian *Web Service* mengirimkan data seluruh laporan keluhan yang diminta dalam *format* JSON kepada *TimelineFragment*. Setelah data diterima, selanjutnya *TimelineFragment* memuat data tersebut ke *fragment_timeline*.

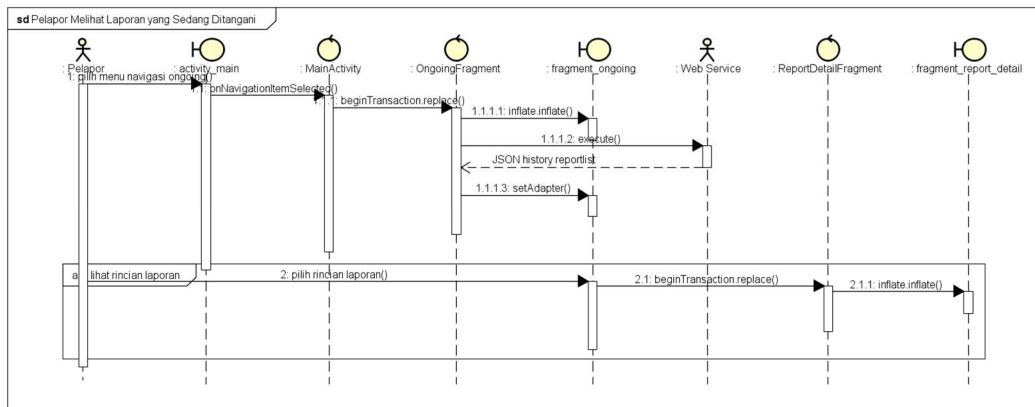


Gambar 5.3 Sequence Diagram Melihat Laporan Lini Masa

Gambar 5.3 juga menunjukkan alur alternatif ketika pelapor memilih fungsi untuk melihat rincian laporan pada *fragment_timeline*. Selanjutnya *fragment* tersebut akan menjalankan fungsi untuk mengganti *fragment* yang sedang dimuat menjadi *ReportDetailFragment* dan dilanjutkan dengan memuat tampilan *fragment_report_detail* yang menampilkan data lengkap mengenai laporan keluhan yang telah dipilih.

5.2.1.3 Sequence Diagram Melihat Laporan yang Sedang Ditangani

Sequence diagram melihat laporan yang sedang ditangani mengacu pada *use case* dengan kode UCP-05 dan dapat dilihat pada Gambar 5.4. Alur ini diawali dengan pelapor memilih fungsi laporan yang sedang ditangani pada menu navigasi. Kemudian setelah itu tampilan *activity_main* akan menjalankan fungsi *onNavigationItemSelected()* pada *MainActivity* untuk mengganti *fragment* yang sedang dimuat di *MainActivity* menjadi *OngoingFragment* lalu *fragment* tersebut memuat tampilan *fragment_ongoing* yang berisi daftar laporan keluhan yang telah dilaporkan oleh pelapor dan sedang ditindak-lanjuti oleh *operator*. Selanjutnya *OngoingFragment* menjalankan fungsi *execute()* untuk meminta data laporan keluhan yang telah dilaporkan oleh pelapor dan sedang ditindak-lanjuti kepada *Web Service* yang berperan sebagai perantara dengan basis data SIMA, kemudian *Web Service* mengirimkan data seluruh laporan keluhan yang diminta dalam *format* JSON kepada *OngoingFragment*. Setelah data diterima, selanjutnya *OngoingFragment* memuat data tersebut ke *fragment_ongoing*.

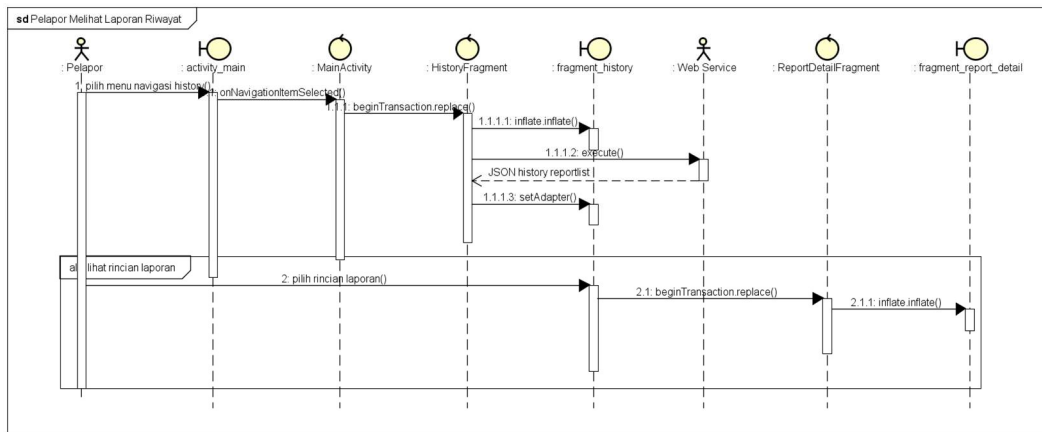


Gambar 5.4 Sequence Diagram Melihat Laporan yang Sedang Ditangani

Gambar 5.4 juga menunjukkan alur alternatif ketika pelapor memilih fungsi untuk melihat rincian laporan pada *fragment_ongoing*. Selanjutnya *fragment* tersebut akan menjalankan fungsi untuk mengganti *fragment* yang sedang dimuat menjadi *ReportDetailFragment* dan dilanjutkan dengan memuat tampilan *fragment_report_detail* yang menampilkan data lengkap mengenai laporan keluhan yang telah dipilih.

5.2.1.4 Sequence Diagram Melihat Laporan Riwayat

Sequence diagram melihat laporan riwayat mengacu pada *use case* dengan kode UCP-06 dan dapat dilihat pada Gambar 5.5. Alur ini diawali dengan pelapor memilih fungsi history pada menu navigasi. Kemudian setelah itu tampilan *activity_main* akan menjalankan fungsi *onNavigationItemSelectedListener()* pada *MainActivity* untuk mengganti *fragment* yang sedang dimuat di *MainActivity* menjadi *HistoryFragment* lalu *fragment* tersebut memuat tampilan *fragment_history* yang berisi daftar laporan keluhan yang telah dilaporkan oleh pelapor dan telah diselesaikan atau ditolak oleh *operator*. Selanjutnya *HistoryFragment* menjalankan fungsi *execute()* untuk meminta data laporan keluhan yang telah dilaporkan oleh pelapor dan telah diselesaikan atau ditolak kepada *Web Service* yang berperan sebagai perantara dengan basis data SIMA, kemudian *Web Service* mengirimkan data seluruh laporan keluhan yang diminta dalam *format* JSON kepada *HistoryFragment*. Setelah data diterima, selanjutnya *HistoryFragment* memuat data tersebut ke *fragment_history*.



Gambar 5.5 Sequence Diagram Melihat Laporan Riwayat

Gambar 5.5 juga menunjukkan alur alternatif ketika pelapor memilih fungsi untuk melihat rincian laporan pada *fragment_history*. Selanjutnya *fragment* tersebut akan menjalankan fungsi untuk mengganti *fragment* yang sedang dimuat menjadi *ReportDetailFragment* dan dilanjutkan dengan memuat tampilan *fragment_report_detail* yang menampilkan data lengkap mengenai laporan keluhan yang telah dipilih.

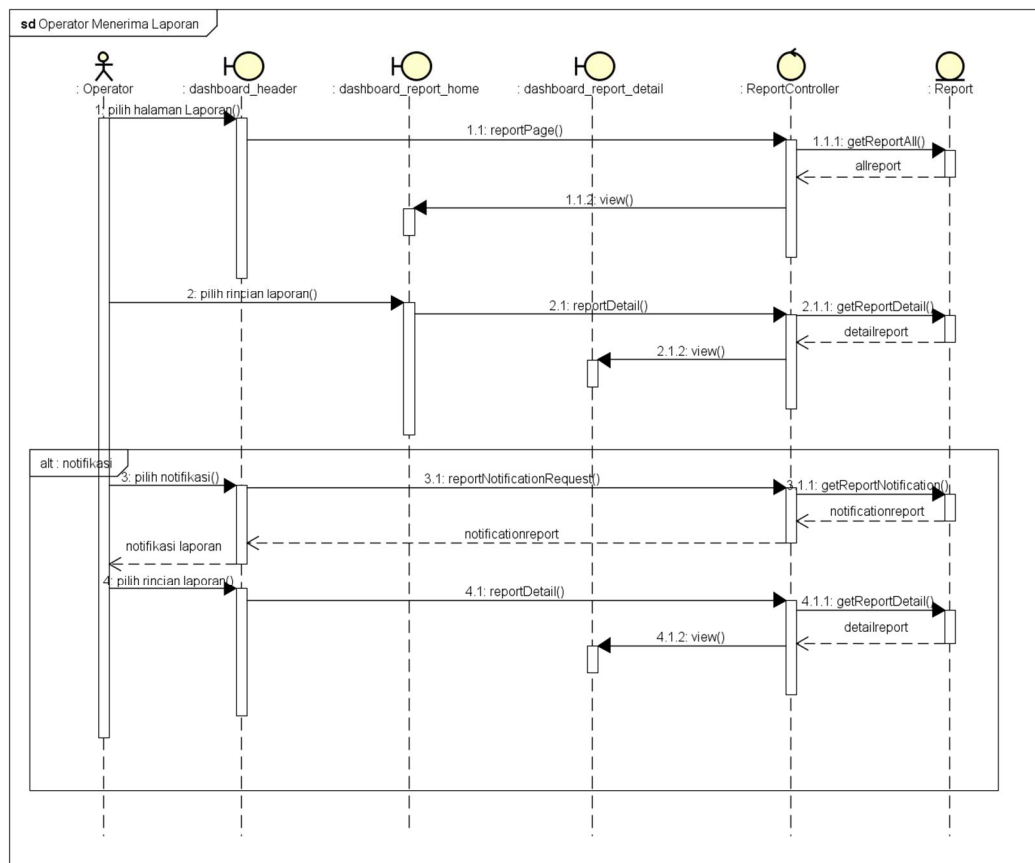
5.2.2 Perancangan *Sequence Diagram* Operator

Pada bagian ini merupakan penggambaran *sequence diagram* dengan aktor *operator*. *Sequence diagram* yang digambarkan pada bagian ini mengacu kepada *use case diagram* uc-operator.

5.2.2.1 *Sequence Diagram* Menerima Laporan

Sequence diagram menerima laporan mengacu pada *use case* dengan kode UCO-03 dan dapat dilihat pada Gambar 5.6. Alur ini diawali dengan operator memilih fungsi untuk menampilkan halaman Laporan. Halaman *dashboard_header* akan menjalankan fungsi *reportPage()* pada *controller* *ReportController*. Untuk mendapatkan seluruh data laporan, fungsi *reportPage()* akan menjalankan fungsi *getReportAll()* terlebih dahulu pada *model* *Report* lalu *model* tersebut akan mengembalikan data *allreport* kepada fungsi *reportPage()*. Setelah itu fungsi *reportPage()* akan memuat halaman *dashboard_report_home* beserta dengan memuat data *allreport* yang telah didapatkan.

Selanjutnya operator dapat memilih fungsi untuk menampilkan rincian dari laporan tertentu pada halaman *dashboard_report_home* yang mana kemudian akan menjalankan fungsi *reportDetail()* pada *controller* *ReportController*. Untuk mendapatkan data lengkap dari laporan tersebut, fungsi *reportDetail()* akan menjalankan fungsi *getReportDetail()* terlebih dahulu pada *Report* lalu *model* tersebut akan mengembalikan data *detailreport* kepada fungsi *reportDetail()*. Setelah itu fungsi *reportDetail()* akan memuat halaman *dashboard_report_detail* beserta dengan memuat data *detailreport* yang telah didapatkan.

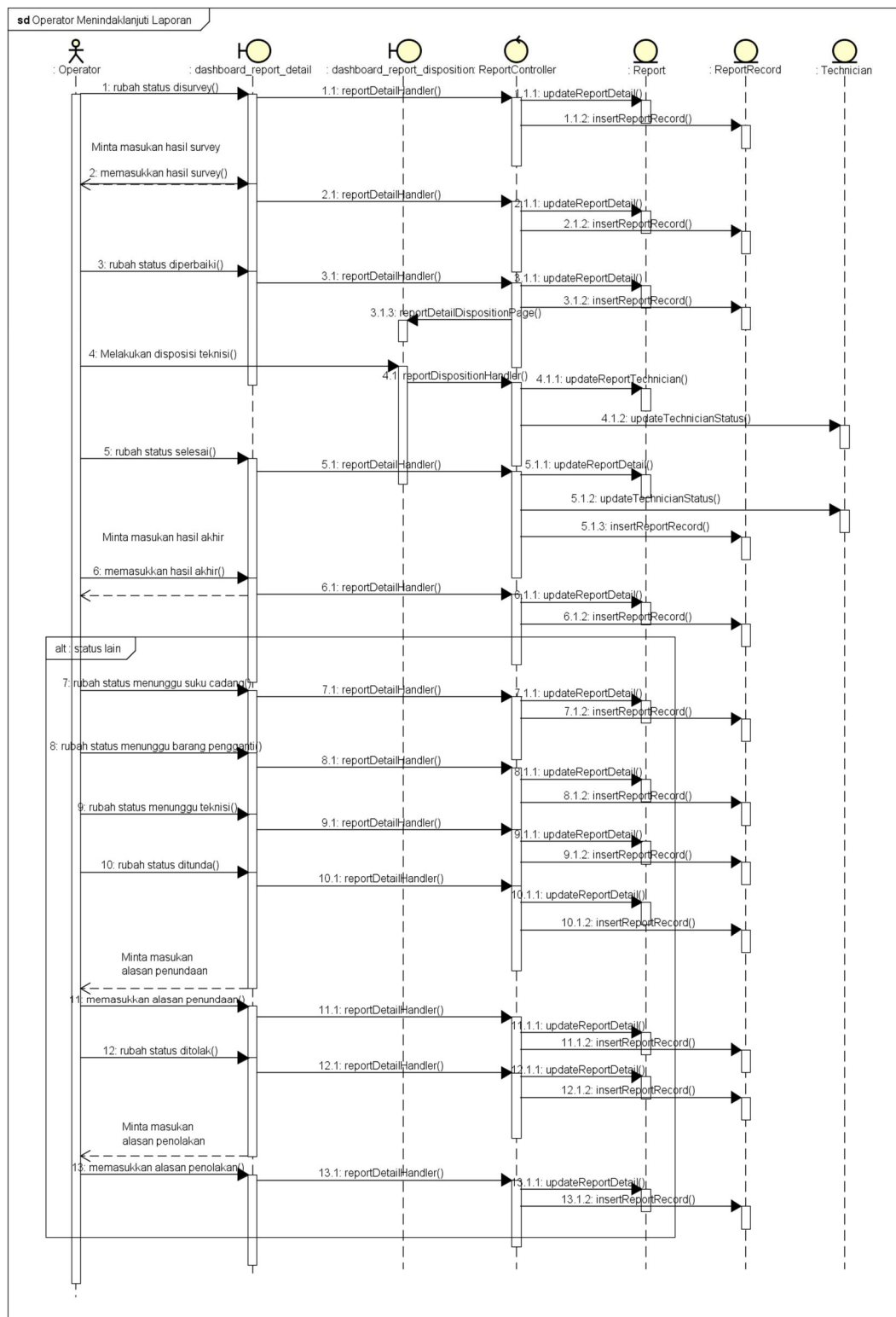


Gambar 5.6 Sequence Diagram Menerima Laporan

Gambar 5.6 juga menunjukkan alur alternatif ketika operator memilih fungsi untuk melihat notifikasi masuk maka `dashboard_header` akan memanggil fungsi `reportNotificationRequest()` pada *controller* `ReportController`. Untuk mendapatkan data laporan yang akan ditampilkan pada notifikasi, fungsi `reportNotificationRequest()` akan menjalankan fungsi `getReportNotification` terlebih dahulu pada `Report` lalu model tersebut akan mengembalikan data `notificationreport` kepada fungsi `reportNotificationRequest()`. Setelah itu fungsi tersebut akan mengirimkan data `notificationreport` ke halaman `dashboard_header` untuk ditampilkan kepada operator. Setelah ditampilkan, operator dapat memilih laporan tertentu yang ada pada notifikasi untuk ditampilkan rincian dari laporan tersebut.

5.2.2.2 Sequence Diagram Menindaklanjuti Laporan

Sequence diagram menindaklanjuti laporan mengacu pada *use case* dengan kode UCO-04 Menindaklanjuti Laporan dan dapat dilihat pada Gambar 5.7.



Gambar 5.7 Sequence Diagram Menindaklanjuti Laporan

Alur ini diawali dengan operator merubah status pelaporan pada halaman dashboard_report_detail menjadi sedang disurvey, kemudian halaman tersebut akan menjalankan fungsi reportDetailHandler() pada *controller* ReportController untuk menangkap perubahan data dan status pelaporan. Lalu untuk menyimpan

perubahan data tersebut *controller* ReportController menjalankan fungsi `updateReportDetail()` pada *model* Report. Setelah itu halaman `dashboard_report_detail` akan lanjut meminta masukan hasil survey kepada operator. Selanjutnya ketika operator telah memasukkan hasil survey dan memilih fungsi *submit*, halaman `dashboard_report_detail` akan memanggil fungsi `reportDetailHandler()` pada *controller* ReportController untuk menangkap perubahan data yang kemudian untuk menyimpan perubahan data tersebut *controller* ReportController menjalankan fungsi `updateReportDetail()` pada *model* Report.

Selanjutnya ketika operator melakukan perubahan status pelaporan menjadi diperbaiki, maka halaman `dashboard_report_detail` akan menjalankan fungsi `reportDetailHandler()` pada *controller* ReportController kemudian menjalankan fungsi `reportDetailDispositionPage()` untuk menampilkan halaman `dashboard_report_disposition`. Dilanjutkan dengan operator melakukan disposisi terhadap teknisi yang tersedia lalu halaman `dashboard_report_disposition` memanggil fungsi `reportDispositionHandler` pada *controller* ReportController untuk menangkap data laporan yang dikerjakan beserta dengan teknisi yang didisposisi dan diteruskan dengan memanggil fungsi `updateReportTechnician()` pada *model* Report dan `updateTechnicianStatus()` pada *model* Technician untuk menyimpan perubahan data status laporan dan ketersediaan teknisi.

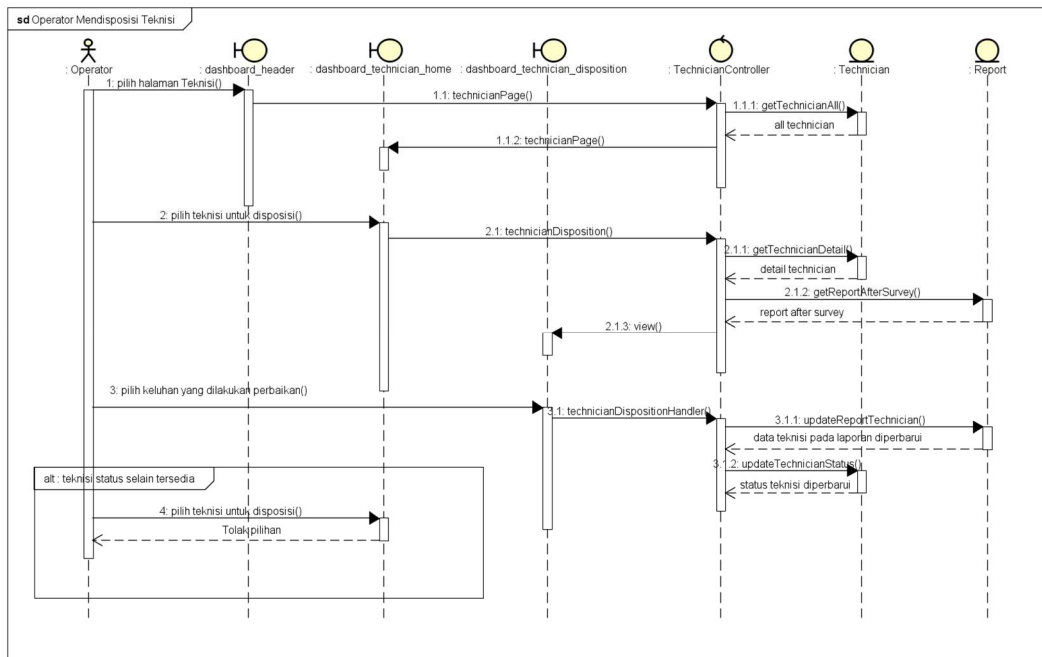
Setelah laporan telah selesai dikerjakan, selanjutnya operator merubah status pelaporan menjadi selesai dan halaman `dashboard_report_detail` akan memanggil fungsi `reportDetailHandler()` pada *controller* ReportController untuk menangkap perubahan data. Untuk menyimpan perubahan status laporan dan status ketersediaan teknisi, fungsi `reportDetailHandler()` akan menjalankan fungsi `updateReportDetail()` pada *model* Report dan `updateTechnicianStatus()` pada *model* Technician. Setelah itu halaman `dashboard_report_detail` akan meminta masukan hasil akhir kepada operator sebagai informasi hasil perbaikan. Operator melanjutkan dengan memasukkan hasil akhir kedalam halaman `dashboard_report_detail` yang kemudian halaman tersebut memanggil fungsi `reportDetailHandler()` pada *controller* ReportController dilanjutkan dengan memanggil fungsi `updateReportDetail()` pada *model* Report untuk menyimpan masukan hasil akhir dari operator.

Pada Gambar 5.7 juga menunjukkan alur alternatif ketika operator merubah status pelaporan menjadi menunggu suku cadang, menunggu barang pengganti dan menunggu teknisi. Setelah operator merubah status maka halaman `dashboard_report_detail` akan memanggil fungsi `reportDetailHandler()` pada *controller* ReportController untuk menangkap perubahan data kemudian dilanjutkan dengan memanggil fungsi `updateReportDetail()` pada *model* Report untuk menyimpan perubahan status pelaporan. Selain itu ketika operator merubah status pelaporan menjadi ditunda atau ditolak maka setelah *model* Report menjalankan fungsi `updateReportDetail()`, halaman `dashboard_report_detail` akan meminta masukan alasan penundaan atau penolakan kepada operator.

5.2.2.3 Sequence Diagram Mendisposisi Teknisi

Sequence diagram mendisposisi teknisi mengacu pada *use case* dengan kode UCO-10 dan dapat dilihat pada Gambar 5.8. Alur ini diawali dengan operator memilih fungsi untuk menampilkan halaman Teknisi. Halaman `dashboard_header` akan menjalankan fungsi `technicianPage()` pada *controller* `TechnicianController`. Untuk mendapatkan seluruh data teknisi, fungsi `technicianPage()` akan menjalankan fungsi `getTechnicianAll()` terlebih dahulu pada *model* `Technician` lalu *model* tersebut akan mengembalikan data `alltechnician` kepada fungsi `technicianPage()`. Setelah itu fungsi `technicianPage()` akan memuat halaman `dashboard_technician_home` beserta dengan memuat data `alltechnician` yang telah didapatkan.

Selanjutnya operator dapat memilih teknisi yang ditampilkan pada halaman `dashboard_technician_home` untuk diberikan disposisi. Setelah memilih teknisi yang diberikan disposisi, halaman `dashboard_technician_home` memanggil fungsi `technicianDisposition()` pada *controller* `TechnicianController`. Kemudian untuk mendapatkan data `detailtechnician` atau data lengkap mengenai teknisi yang akan diberikan disposisi dan data `reportaftersurvey` atau data daftar laporan keluhan yang dapat mulai dikerjakan, `technicianDisposition()` menjalankan fungsi `getTechnicianDetail()` pada *model* `Technician` dan `getReportAfterSurvey()` pada *model* `Report`. Lalu fungsi `technicianDisposition` menampilkan halaman `dashboard_technician_disposition` dan memuat data `detailtechnician` dan `reportaftersurvey`. Setelah itu operator melanjutkan dengan memilih salah satu laporan keluhan yang harus dikerjakan oleh teknisi pada halaman `dashboard_technician_disposition`, kemudian halaman tersebut memanggil fungsi `technicianDispositionHandler()` pada *controller* `TechnicianController` untuk menangkap data disposisi dan dilanjutkan dengan menjalankan fungsi `updateReportTechnician()` dan `updateTechnicianStatus()` untuk menyimpan perubahan status laporan dan status ketersediaan teknisi.



Gambar 5.8 Sequence Diagram Mendisposisi Teknisi

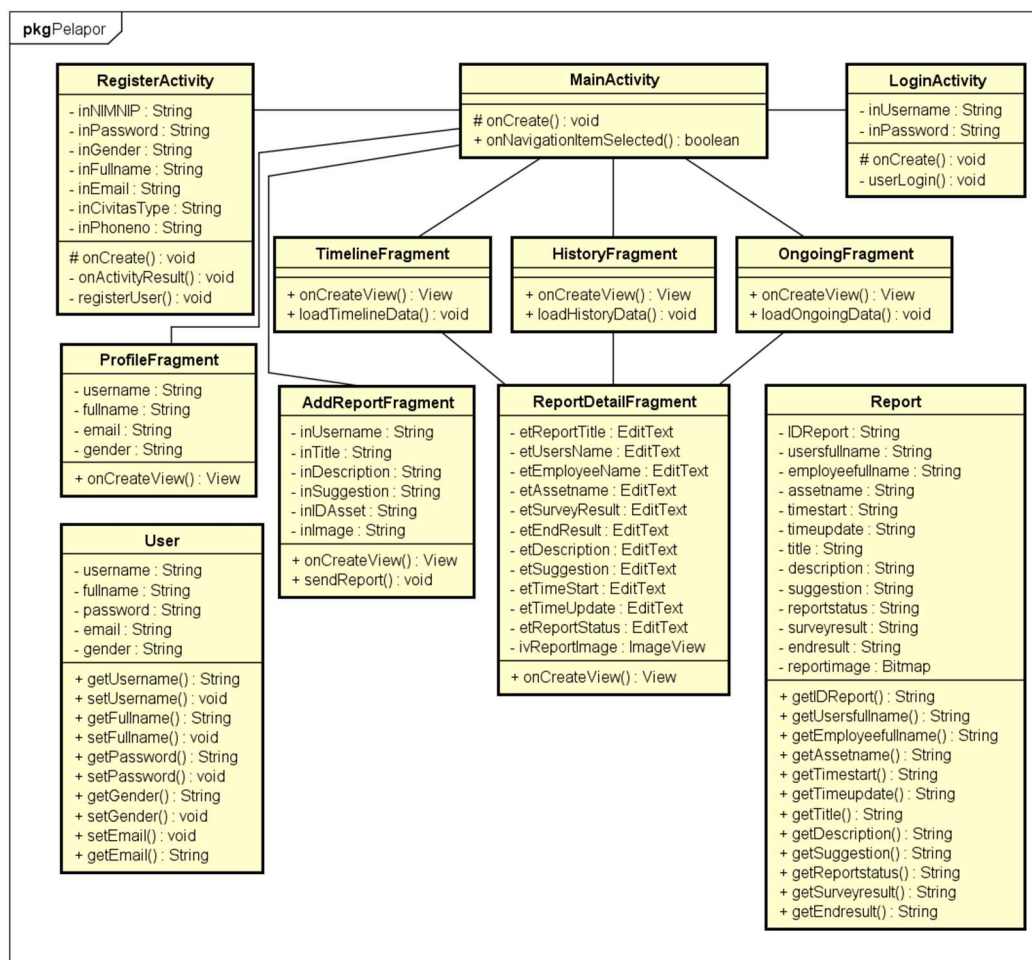
Pada Gambar 5.8 juga menunjukkan alur alternatif ketika operator melakukan disposisi dan memilih teknisi yang status nya selain tersedia, maka halaman `dashboard_technician_home` akan menolak pilihan dari operator sehingga operator hanya dapat melakukan disposisi teknisi yang statusnya tersedia saja.

5.3 Perancangan Class Diagram

Pada bagian ini digambarkan *class* atau *object* apa saja yang saling berkaitan, berhubungan dan saling bertukar informasi di dalam Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Pada perancangan *class diagram* berikut ini akan dibagi menjadi dua bagian sesuai aktor yang telah diidentifikasi pada tahap analisis dan perancangan sequence diagram, yaitu perancangan *class diagram* pelapor dan perancangan *class diagram* operator. Selain itu, pada bagian ini juga akan digambarkan *class diagram* yang digunakan sebagai *domain model* untuk pemodelan skema basis data.

5.3.1 Perancangan Class Diagram Pelapor

Bagian ini merupakan penggambaran *class diagram* dengan aktor pelapor. *Class diagram* yang digambarkan pada bagian ini dijadikan sebagai acuan dari kelas-kelas yang akan digunakan dalam pembangunan aplikasi *android* Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana untuk aktor pelapor.



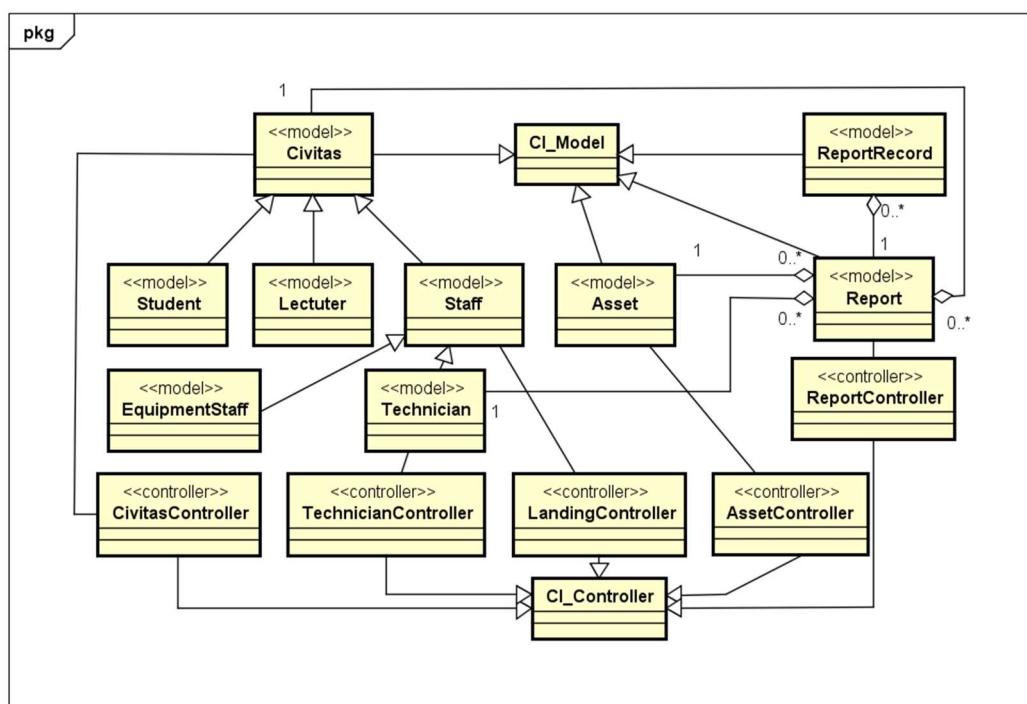
Gambar 5.9 Class Diagram Aplikasi Mobile Pelapor

Gambar 5.9 menjelaskan bahwa aplikasi *android* Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana dibangun menggunakan komponen pemrograman aplikasi *android* dengan bahasa pemrograman *Java*. Dalam penerapannya kelas-kelas yang dijelaskan dapat berupa *Activity*, *Fragment* dan *Java Class* biasa.

Pada aplikasi *android* Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana kelas *Activity* terdiri dari *LoginActivity*, *RegisterActivity* dan *MainActivity*. *LoginActivity* dan *RegisterActivity* merupakan perwujudan kelas *Activity* untuk aktor *Guest* dari *use case diagram* uc-pelapor, sedangkan *MainActivity* merupakan perwujudan kelas *Activity* untuk aktor *Pelapor* dari *use case diagram* uc-pelapor. *LoginActivity* dan *RegisterActivity* berisi operasi-operasi yang disediakan dan dapat dilakukan oleh aktor *Guest* seperti melakukan login ke dalam menu utama aplikasi *android* dan pendaftaran akun pelapor baru. *MainActivity* berisi menu fungsi yang disediakan dan dapat dilakukan oleh aktor *Pelapor*. Fungsi-fungsi yang disediakan antara lain adalah untuk mengirim laporan diwujudkan dengan kelas *AddReportFragment*, melihat laporan lini masa diwujudkan dengan *TimelineFragment*, melihat laporan yang sedang ditangani diwujudkan dengan *OnGoingFragment*, melihat laporan riwayat diwujudkan

5.3.2 Perancangan *Class Diagram Operator*

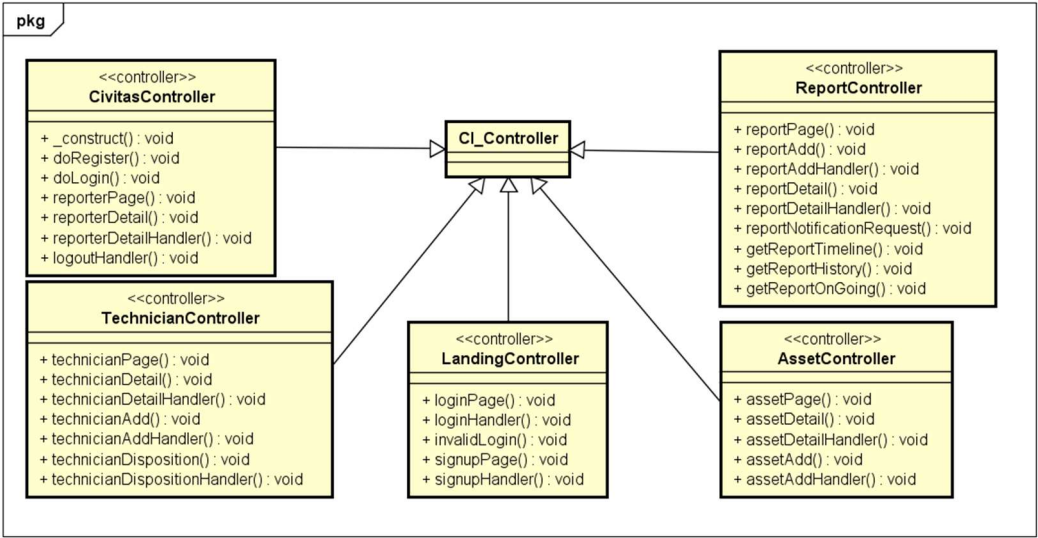
gambarkan relasi kelas-kelas m



Gambar 5.10 Class Diagram Relasi Antar Kelas Website Operator

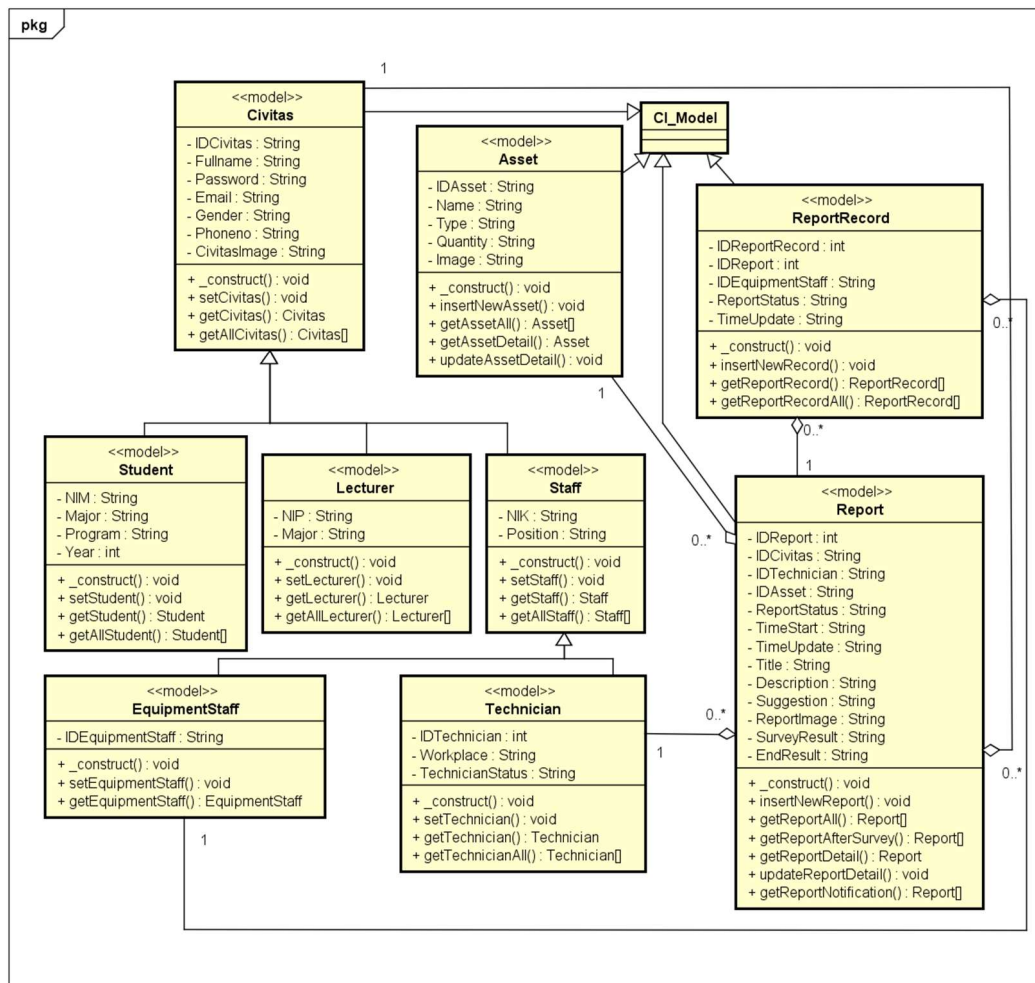
Gambar 5.10 menjelaskan mengenai relasi antar kelas pada *website operator* bahwa *website operator* Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana dibangun menggunakan *framework Codeigniter*. Framework ini didasarkan pada pola perancangan *Model-View-Controller* sehingga dalam penerapannya kelas-kelas akan dibagi menjadi kelas *controller* dan kelas *model*. Kelas-kelas yang akan diimplementasikan pada *website operator* terdiri dari 5 kelas controller yang mewarisi dari kelas *CI_Controller* dan 9 kelas model yang mewarisi dari kelas *CI_Model*. Kelas controller tersebut terdiri dari kelas *CivitasController*, *TechnicianController*, *LandingController*, *AssetController* dan

ReportController. Sedangkan kelas model terdiri dari kelas Civitas, Student, Lecturer, Staff, EquipmentStaff, Technician, Asset, Report dan ReportRecord.



Gambar 5.11 Class Diagram Relasi Antar Kelas Controller

Berikut pada Gambar 5.11 merupakan *class diagram* yang menjelaskan relasi dari kelas-kelas *controller*. *CivitasController* merupakan kelas *controller* yang digunakan untuk menampilkan informasi daftar civitas akademik yang terdaftar sebagai pelapor kepada *operator* serta sebagai *web service* untuk melakukan login dan pendaftaran akun pelapor yang dapat diakses oleh aplikasi *android* Sistem Informasi Manajemen Aset atau aplikasi lain. Kelas *LandingController* merupakan kelas *controller* yang digunakan untuk menampilkan halaman login dan halaman pendaftaran akun operator, serta fungsi-fungsi untuk mengolah data login dan pendaftaran. Kelas *AssetController* merupakan kelas *controller* yang digunakan untuk menampilkan informasi daftar asset yang telah dibuat, membuat serta mengolah pembaharuan data dari asset tersebut. Kelas *TechnicianController* merupakan kelas *controller* yang digunakan untuk menampilkan informasi daftar teknisi yang terdaftar, mendaftarkan dan mengolah pembaharuan data dari teknisi serta menyediakan fungsi untuk melakukan disposisi teknisi. Sedangkan kelas *ReportController* merupakan kelas *controller* yang digunakan untuk menampilkan informasi daftar keluhan yang telah dilaporkan, membuat serta mengolah pembaharuan data laporan keluhan sarana prasarana. Selain itu kelas *ReportController* juga memiliki fungsi sebagai *web service* untuk menyediakan data laporan lini masa, yang sedang ditangani dan riwayat yang dapat diakses oleh aplikasi *android* Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana atau aplikasi lain.



Gambar 5.12 Class Diagram Relasi Antar Kelas Model

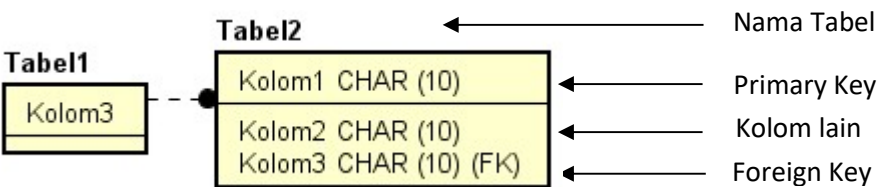
Berikut pada Gambar 5.12 merupakan *class diagram* yang menjelaskan relasi dari kelas-kelas *model*. Kelas Civitas merupakan kelas *model* yang menjadi representasi dari entitas civitas akademik secara keseluruhan baik pelapor maupun *operator* dan digunakan untuk operasi yang berhubungan dengan proses membuat data civitas baru dan mendapatkan data seluruh civitas atau civitas tertentu. Kelas Student merupakan kelas *model* turunan dari kelas Civitas yang menjadi representasi dari entitas mahasiswa sebagai pelapor dan digunakan untuk operasi yang berhubungan dengan proses membuat data mahasiswa baru dan mendapatkan data seluruh mahasiswa atau mahasiswa tertentu. Kelas Lecturer merupakan kelas *model* turunan dari kelas Civitas yang menjadi representasi dari entitas dosen sebagai pelapor dan digunakan untuk operasi yang berhubungan dengan proses membuat data dosen baru dan mendapatkan data seluruh dosen atau dosen tertentu. Kelas Staff merupakan kelas *model* turunan dari kelas Civitas yang menjadi representasi dari entitas pegawai sebagai pelapor dan digunakan untuk operasi yang berhubungan dengan proses membuat data pegawai baru dan mendapatkan data seluruh pegawai atau pegawai tertentu. Kelas EquipmentStaff merupakan kelas *model* turunan dari kelas Staff yang menjadi representasi dari entitas pegawai perlengkapan sebagai *operator* dan

digunakan untuk operasi yang berhubungan dengan proses membuat data pegawai baru dan mendapatkan data seluruh pegawai perlengkapan atau pegawai perlengkapan tertentu. Kelas Technician merupakan kelas *model* turunan dari kelas Staff yang menjadi representasi dari entitas teknisi dan digunakan untuk operasi yang berhubungan dengan proses membuat data teknisi baru dan mendapatkan data seluruh teknisi atau teknisi tertentu.

Kelas Asset merupakan kelas *model* yang menjadi representasi dari entitas sarana prasarana dan digunakan untuk operasi yang berhubungan dengan proses membuat data aset baru dan mendapatkan data seluruh aset atau aset tertentu. Kelas Report merupakan kelas *model* yang menjadi representasi dari entitas laporan dan digunakan untuk operasi yang berhubungan dengan proses membuat laporan baru, kemudian mendapatkan data seluruh laporan, seluruh data laporan yang telah disurvey, data laporan tertentu, memperbarui data laporan dan mendapatkan data laporan terbaru untuk keperluan notifikasi. Sedangkan kelas ReportRecord merupakan kelas *model* yang menjadi representasi dari entitas rekam laporan untuk melacak perubahan status laporan yang memiliki operasi yang berhubungan dengan proses membuat rekam laporan baru dan mendapatkan data rekam laporan.

5.4 Perancangan Skema Basis Data

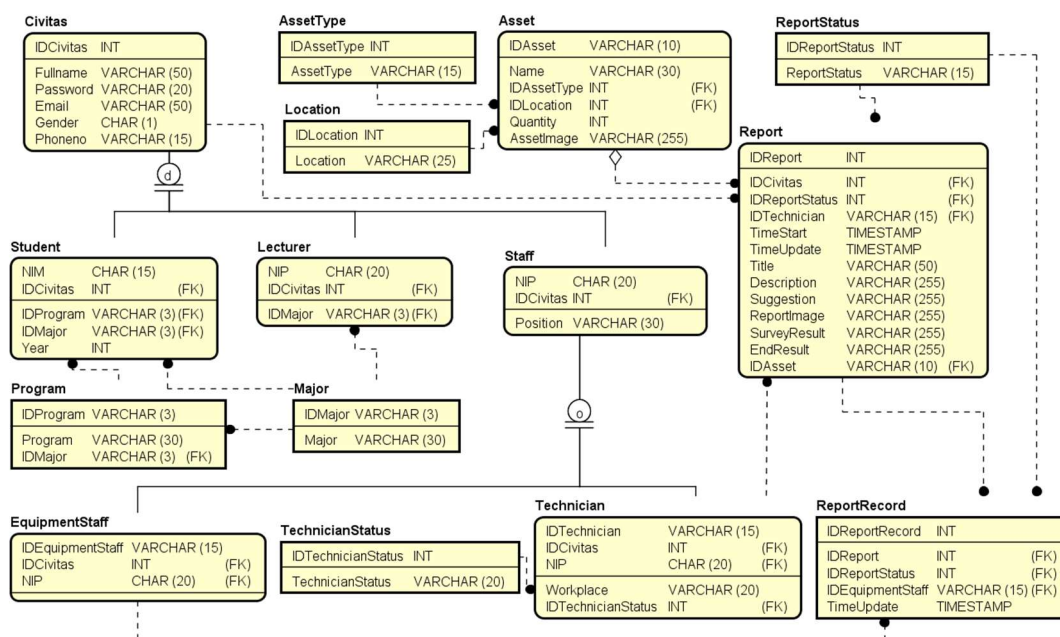
Bagian ini merupakan penggambaran skema basis data yang digunakan sebagai acuan untuk membangun basis data dari Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Perancangan skema basis data pada bagian ini menggunakan alat bantu perancangan perancangan Astah Professional yang mana menyediakan bentuk persegi panjang dengan memberikan garis pembatas didalamnya dalam menggambarkan sebuah tabel. Bagian di atas garis pembatas dari suatu tabel merupakan penjelasan dari atribut primary key dari tabel tersebut. Berikutnya bagian di bawah garis pembatas merupakan penjelasan dari atribut lain dari tabel tersebut. Apabila dari atribut-atribut tersebut terdapat imbuhan (FK) di akhirannya maka kolom tersebut merupakan atribut *foreign key*, yang mana seperti pada kolom3 dari Tabel2 mereferensi kepada Kolom3 dari Tabel1. Gambar 5.13 merupakan contoh penggambaran *physical data model* menggunakan Astah Professional.



Gambar 5.13 Contoh penggambaran physical data model

Pada Gambar 5.14 menunjukkan perancangan skema basis data menggunakan *Physical Data Model* (PDM) yang mengacu pada *class diagram* relasi antar kelas model pada Gambar 5.12. Dari hasil pemetaan *class diagram* relasi antar kelas model ke PDM dan melalui proses normalisasi didapatkan 15 tabel yang terdiri

dari tabel *Civitas*, *Student*, *Lecturer*, *Staff*, *EquipmentStaff*, *Technician*, *TechnicianStatus*, *Program*, *Major*, *Asset*, *AssetType*, *Location*, *Report*, *ReportStatus*, *ReportRecord*.



Gambar 5.14 Physical Data Model Basis Data SIMA

Tabel *Student*, *Lecturer* dan *Staff* pada Gambar 5.14 merupakan tabel turunan yang bersifat *disjoint* dari tabel *Civitas*, yang berarti satu data civitas dapat menjadi anggota dari salah satu data pada tabel *Student*, *Lecturer* atau *Staff*. Sedangkan tabel *EquipmentStaff* dan *Technician* merupakan tabel turunan yang bersifat *overlapping* dari tabel *Staff*, yang berarti satu data staff dapat menjadi anggota dari data pada tabel *EquipmentStaff* sekaligus data pada tabel *Technician*.

5.5 Perancangan Antarmuka Pengguna

Pada bagian ini akan ditunjukkan perancangan antarmuka pengguna yang dilakukan untuk menggambarkan tampilan dari sistem yang akan dibangun. Berdasarkan jenis aktor yang telah teridentifikasi pada tahap sebelumnya, perancangan antarmuka pengguna akan dibagi menjadi 2 bagian yaitu perancangan antarmuka untuk pengguna pelapor dan perancangan antarmuka untuk pengguna *operator*.

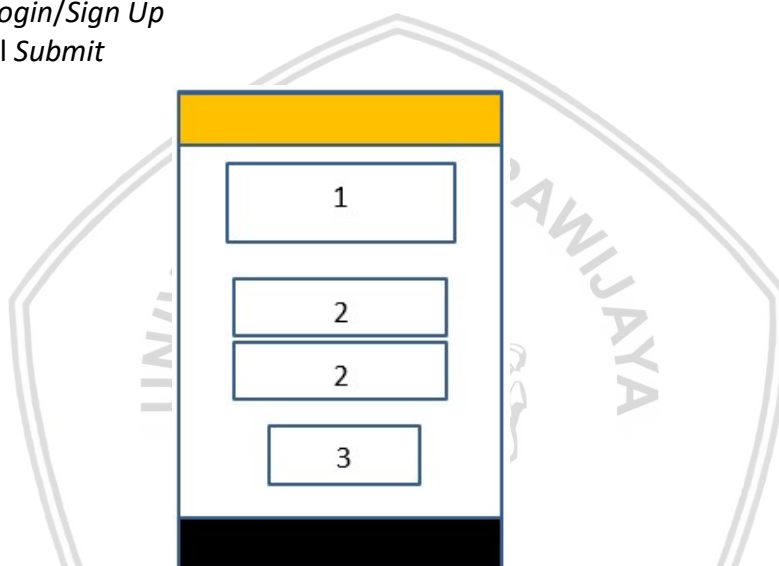
5.5.1 Perancangan Antarmuka Pengguna Pelapor

Bagian ini menunjukkan tampilan dari aplikasi *android* yang akan dibangun untuk aktor pelapor. Perancangan antarmuka pengguna pelapor berikut akan dibagi lagi menjadi rancangan halaman login dan sign up, halaman membuat laporan baru, halaman lini masa, yang sedang ditangani, riwayat laporan serta halaman detail laporan.

5.5.1.1 Rancangan Halaman Login dan Sign Up Pelapor

Berikut pada Gambar 5.15 merupakan rancangan antarmuka pelapor untuk halaman *login* dan *sign up*. Keterangan dari Gambar 5.15 antara lain adalah:

1. Logo Aplikasi SIMA
2. Form *Login/Sign Up*
3. Tombol *Submit*



Gambar 5.15 Rancangan Halaman Login dan Sign Up Pelapor

5.5.1.2 Rancangan Halaman Laporan Baru

Berikut pada Gambar 5.16 merupakan rancangan antarmuka pelapor untuk halaman laporan baru. Keterangan dari Gambar 5.16 antara lain adalah:

1. Judul Halaman
2. Form Data Laporan
3. Tombol *Scan QR Code*
4. Tombol Kamera
5. Tombol *Submit*

Gambar 5.16 Rancangan Halaman Laporan Baru

5.5.1.3 Rancangan Halaman Laporan Lini Masa, Sedang Berjalan dan Riwayat

Berikut pada Gambar 5.17 merupakan rancangan antarmuka pelapor untuk halaman laporan lini masa, laporan yang sedang berjalan dan laporan riwayat. Keterangan dari Gambar 5.17 antara lain adalah:

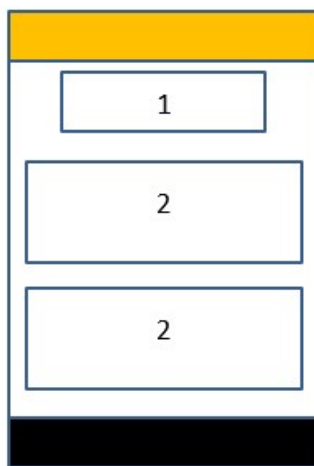
1. Judul Halaman
2. Daftar Laporan
3. Foto Laporan

Gambar 5.17 Rancangan Halaman Lini Masa, yang Sedang Ditnagani dan Laporan Riwayat

5.5.1.4 Rancangan Halaman Rincian Laporan

Berikut pada Gambar 5.18 merupakan rancangan antarmuka pelapor untuk halaman rincian laporan. Keterangan dari Gambar 5.18 antara lain adalah:

1. Judul Halaman
2. Form Isi Rincian Laporan



Gambar 5.18 Rancangan Halaman Rincian Laporan

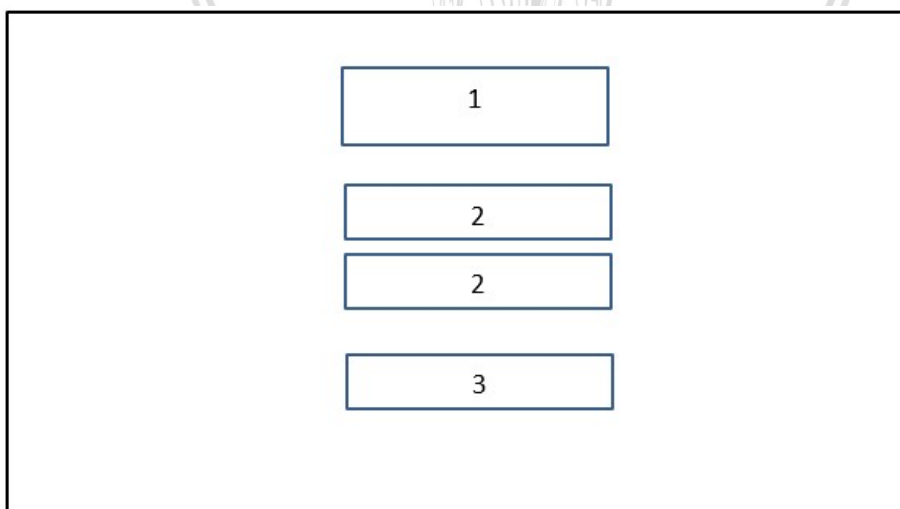
5.5.2 Perancangan Antarmuka Pengguna Operator

Bagian ini menunjukkan tampilan dari *website operator* yang akan dibangun untuk aktor *operator*. Perancangan antarmuka pengguna *operator* berikut akan dibagi lagi menjadi rancangan halaman login dan sign up, halaman daftar laporan, aset dan teknisi pelapor serta halaman detail laporan, aset dan teknisi pelapor.

5.5.2.1 Rancangan Halaman Login dan Sign Up Operator

Berikut pada Gambar 5.19 merupakan rancangan antarmuka *operator* untuk halaman *login* dan *sign up*. Keterangan dari Gambar 5.19 antara lain adalah:

1. Logo Website Operator SIMA
2. Form Login/Sign Up
3. Tombol Submit

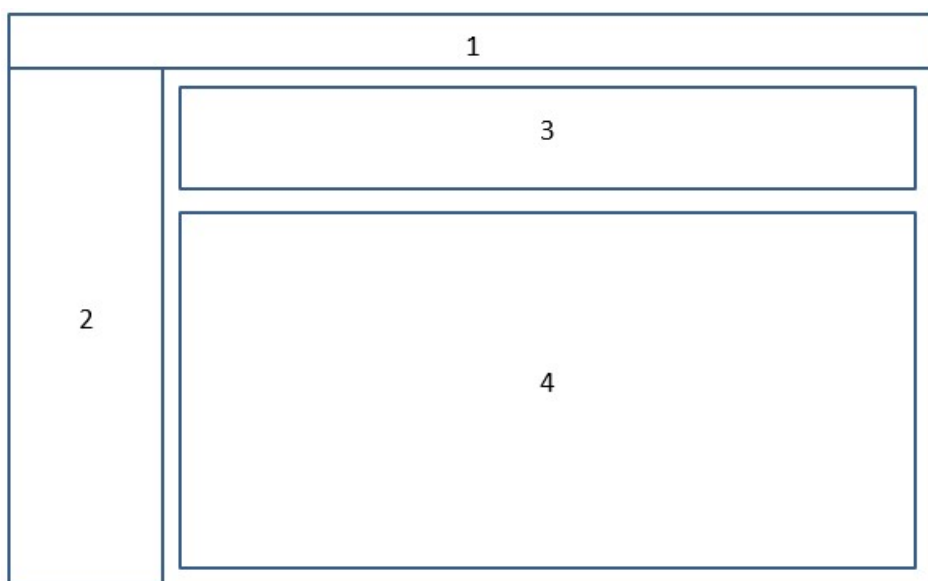


Gambar 5.19 Rancangan Halaman Login dan Sign Up Operator

5.5.2.2 Rancangan Halaman Daftar Laporan, Aset, Teknisi dan Pelapor

Berikut pada Gambar 5.20 merupakan rancangan antarmuka *operator* untuk halaman daftar laporan, aset, teknisi dan pelapor. Keterangan dari Gambar 5.20 antara lain adalah:

1. Header Notifikasi
2. Menu Navigasi
3. Judul Halaman dan Menu Halaman
4. Daftar Data Laporan/Aset/Teknisi/Pelapor

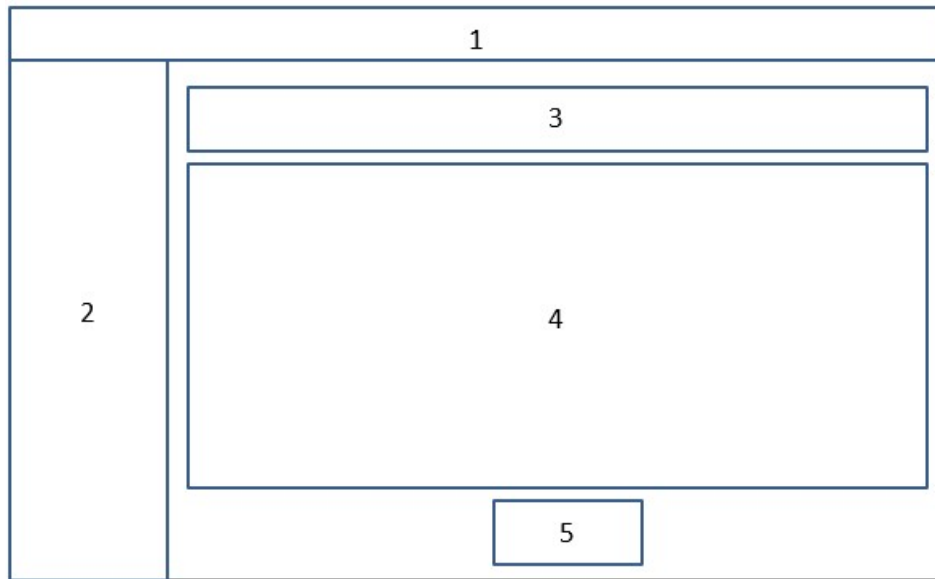


Gambar 5.20 Rancangan Halaman Daftar Laporan, Aset, Teknisi dan Pelapor

5.5.2.3 Rancangan Halaman Rincian Laporan, Aset, Teknisi dan Pelapor

Berikut pada Gambar 5.21 merupakan rancangan antarmuka *operator* untuk halaman rincian laporan, aset, teknisi dan pelapor. Keterangan dari Gambar 5.21 antara lain adalah:

1. Header Notifikasi
2. Menu Navigasi
3. Judul Halaman
4. Form Data Detail Laporan/Aset/Teknisi/Pelapor
5. Tombol Submit

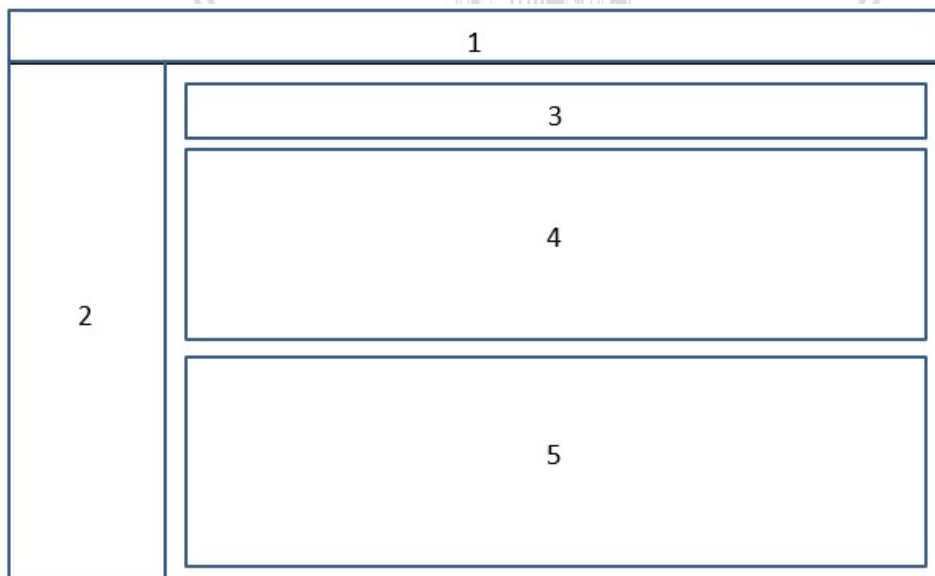


Gambar 5.21 Rancangan Halaman Rincian Laporan, Aset, Teknisi dan Pelapor

5.5.2.4 Rancangan Halaman Disposisi Teknisi

Berikut pada Gambar 5.22 merupakan rancangan antarmuka *operator* untuk halaman disposisi teknisi. Keterangan dari Gambar 5.22 antara lain adalah:

1. Header Notifikasi
2. Menu Navigasi
3. Judul Halaman
4. Form Data Laporan
5. Tabel Daftar Teknisi



Gambar 5.22 Rancangan Halaman Disposisi Teknisi

5.6 Perancangan Algoritme

Pada perancangan algoritme dijelaskan mengenai algoritme yang akan diterapkan untuk implementasi Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Berdasarkan jenis aktor yang telah teridentifikasi pada tahap sebelumnya, perancangan algoritme akan dibagi menjadi 2 bagian yaitu perancangan algoritme untuk pengguna pelapor dan perancangan algoritme untuk pengguna *operator*

5.6.1 Perancangan Algoritme Pelapor

Bagian ini menjelaskan perancangan algoritme dari aplikasi *android* yang digunakan oleh pelapor. Perancangan algoritme yang dijelaskan pada bagian ini mencakup algoritme dari proses pengiriman laporan oleh pelapor dan proses untuk melihat laporan.

5.6.1.1 Algoritme Mengirim Laporan

Dalam proses pengiriman laporan pada aplikasi *android* pelapor terdapat beberapa tahap yang dilakukan oleh sistem. Pada Tabel 5.1 menjelaskan mengenai perancangan algoritme dari proses pengiriman laporan.

Tabel 5.1 Perancangan Algoritme Mengirim Laporan

No	Pseudocode
1	Mulai
2	Jika judul laporan atau kode aset atau deskripsi laporan atau saran tindakan belum terisi
3	Maka keluarkan error
4	Akhir kondisi
5	Jalankan fungsi kirim laporan
6	Mengambil data NIM pelapor yang sedang login
7	Menyimpan data NIM dan data laporan ke dalam variable parameter
8	Buka koneksi dengan alamat web service menggunakan metode POST
9	Kirim data variable parameter ke alamat web service
10	Jika mendapatkan kode respon dari web service
11	Maka simpan kode respon dari web service
12	Akhir kondisi
13	Simpan respon ke dalam objek JSON
14	Simpan objek JSON ke dalam instansiasi objek Report
15	Akhir fungsi kirim laporan
16	Kembali ke halaman menu
17	Selesai

5.6.1.2 Algoritme Melihat Laporan

Dalam proses untuk melihat laporan pada aplikasi *android* pelapor yang mencakup melihat laporan lini masa, laporan yang sedang ditangani dan laporan riwayat terdapat beberapa tahap yang dilakukan oleh sistem. Pada Tabel 5.2 menjelaskan mengenai perancangan algoritme dari proses melihat laporan.

Tabel 5.2 Perancangan Algoritme Melihat Laporan

No	Pseudocode
1	Mulai
2	Buka koneksi dengan alamat url web service untuk mengakses laporan lini masa, sedang ditangani atau riwayat
3	Simpan data respon dari web service ke dalam objek JSON
4	Ulangi sampai seluruh isi JSON
5	Simpan objek JSON ke dalam instansiasi objek Report
6	Tambahkan objek Report ke dalam array Report
7	Selesai perulangan
8	Tampilkan seluruh data array Report
9	Selesai

5.6.2 Perancangan Algoritme Operator

Bagian ini menjelaskan perancangan algoritme dari *website operator* yang digunakan oleh *operator*. Perancangan algoritme yang dijelaskan pada bagian ini mencakup algoritme mengenai proses menerima laporan dari pelapor dan proses menindaklanjuti laporan.

5.6.2.1 Algoritme Menerima Laporan

Dalam proses menerima laporan pada *website operator* terdapat beberapa tahap yang dilakukan oleh sistem. Pada Tabel 5.3 menjelaskan mengenai perancangan algoritme dari proses pengiriman laporan.

Tabel 5.3 Perancangan Algoritme Menerima Laporan

No	Pseudocode
1	Mulai
2	Lakukan request get ke kelas controller untuk mendapatkan data laporan terbaru dari kelas model
3	Menyimpan data laporan dari kelas controller ke variabel lolal
4	Ulangi sampai seluruh isi data laporan
5	Perbarui data laporan yang ditampilkan pada fitur notifikasi
6	Selesai perulangan
7	Ulangi request get setiap selang satuan waktu
8	Selesai

5.6.2.2 Algoritme Menindaklanjuti Laporan

Dalam proses menindaklanjuti laporan pada *website operator* terdapat beberapa tahap yang dilakukan oleh sistem. Pada Tabel 5.4 menjelaskan mengenai perancangan algoritme dari proses pengiriman laporan.

Tabel 5.4 Perancangan Algoritme Menindaklanjuti Laporan

No	Pseudocode
1	Mulai
2	Simpan perubahan data laporan yang sedang ditindaklanjuti ke variabel lokal
3	Jika perubahan status laporan menjadi sedang perbaikan
4	Simpan perubahan data laporan ke basis data melalui kelas model
5	Arahkan ke halaman disposisi teknisi
6	Simpan data laporan yang ditindaklanjuti dan teknisi yang didisposisi ke variabel lokal
7	Simpan data teknisi yang didisposisi terhadap laporan ke basis data melalui kelas model
8	Simpan status ketersediaan teknisi menjadi sedang bekerja ke basis data melalui kelas model
9	Kembali ke halaman detail laporan
10	Akhir kondisi
11	Jika perubahan status laporan menjadi selesai
12	Simpan perubahan data laporan ke basis data melalui kelas model
13	Simpan perubahan status ketersediaan teknisi menjadi tersedia ke basis data melalui kelas model
14	Kembali ke halaman detail laporan
15	Akhir kondisi
16	Selain itu
17	Simpan perubahan data laporan ke basis data melalui kelas model
18	Kembali ke halaman detail laporan
19	Akhir kondisi
20	Selesai

BAB 6 IMPLEMENTASI

Implementasi merupakan representasi dari Tahap Implementasi pada *Waterfall Model* yang mana merupakan proses realisasi dari kebutuhan bisnis, kebutuhan sistem dan spesifikasi perancangan menjadi komponen-komponen program, basis data, website atau aplikasi perangkat lunak konkrit melalui implementasi pemrograman. Pembahasan pada bab implementasi ini diuraikan menjadi batasan implementasi, implementasi program dan implementasi antarmuka.

6.1 Batasan Implementasi

Berikut ini merupakan batasan-batasan dalam implementasi Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana FILKOM UB:

1. Aplikasi *Mobile* pelapor Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana dirancang dan dibangun menggunakan bahasa pemrograman Java dengan platform Android.
2. *Website Operator* Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana dirancang dan dibangun menggunakan bahasa pemrograman PHP dengan menerapkan framework Codeigniter.
3. Manajemen sistem basis data yang dirancang dibangun menggunakan Relational Database Management System (RDBMS) MySQL.
4. Aplikasi *Mobile* pelapor memiliki fitur utama untuk mengirim laporan, melihat status pelaporan keluhan beserta melihat rincian lengkap status pelaporan keluhan.
5. *Website Operator* memiliki fitur utama untuk menerima dan melihat laporan, menindaklanjuti laporan keluhan, mengelola barang dan mengelola teknisi.

6.2 Implementasi Program

Pada bagian ini berisi implementasi dari program Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana yang dikembangkan. Pada implementasi program dari Sistem Informasi Manajemen ini terbagi menjadi tiga bagian, yang mana dua bagian awal merupakan pembagian sesuai aktor yang telah diidentifikasi yaitu implementasi aplikasi *mobile* pelapor dan implementasi *website operator* dan bagian terakhir merupakan implementasi dari *web service*.

6.2.1 Implementasi Aplikasi *Mobile* Pelapor

Pada bagian ini merupakan implementasi dari aplikasi *mobile* pelapor. Implementasi program yang akan dijelaskan pada bagian ini terdiri dari penjelasan mengenai implementasi untuk mengirim laporan dan implementasi untuk melihat laporan sesuai dengan kebutuhan yang dijelaskan pada perancangan *sequence diagram* pelapor dan perancangan algoritme pelapor. Pada bagian ini

implementasi untuk melihat laporan yang akan dijelaskan diwakilkan oleh implementasi melihat lini masa laporan.

6.2.1.1 Implementasi Mengirim Laporan

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian perancangan *sequence diagram* mengirim laporan dan perancangan algoritme mengirim laporan yang telah dijelaskan pada Tabel 5.1 Perancangan Algoritme Mengirim Laporan ke dalam wujud kode program pada kelas `AddReportFragment.java`. Implementasi kode program dari mengirim laporan dijelaskan pada Tabel 6.1 berikut.

Tabel 6.1 Implementasi Mengirim Laporan

No	AddReportFragment.java
1	<code>private void sendReport(){</code>
2	<code> //validasi masukan</code>
3	<code> if (TextUtils.isEmpty(inTitle)){</code>
4	<code> editTextTitle.setError("Judul laporan harap diisi");</code>
5	<code> editTextTitle.requestFocus();</code>
6	<code> return;</code>
7	<code> if (TextUtils.isEmpty(inDescription)){</code>
8	<code> editTextDescription.setError("Deskripsi Laporan harap</code>
9	<code> diisi");</code>
10	<code> editTextDescription.requestFocus();</code>
10	<code> return;</code>
11	<code> if (TextUtils.isEmpty(inSuggestion)){</code>
12	<code> editTextSuggestion.setError("Saran Tindakan harap diisi");</code>
13	<code> editTextSuggestion.requestFocus();</code>
14	<code> return; }</code>
15	<code> if (TextUtils.isEmpty(inIDAsset)){</code>
16	<code> editTextIDAsset.setError("Keterangan Aset harap diisi");</code>
17	<code> editTextIDAsset.requestFocus();</code>
18	<code> return;</code>
	<code> //jika masukan sudah terisi</code>
19	<code> class SendReport extends AsyncTask<Void, Void, String> {</code>
20	<code> @Override</code>
21	<code> protected void onPreExecute() {</code>
22	<code> super.onPreExecute();</code>
23	<code> progressBar.setVisibility(View.VISIBLE);}</code>
24	<code> @Override</code>
25	<code> protected String doInBackground(Void... voids) {</code>
26	<code> //membuat objek requesthandler</code>
27	<code> RequestHandler requestHandler = new RequestHandler();</code>

No	AddReportFragment.java
28	//get username
29	User user =
	SharedPreferences.getInstance(activity).getUser();
30	inUsername = user.getUsername();
31	//jika foto laporan tidak ada
32	if (encodedImage == null) {
33	ByteArrayOutputStream baos = new
	ByteArrayOutputStream();
34	Bitmap bitmap =
	BitmapFactory.decodeResource(getResources(),
	R.drawable.noimageavailable);
35	bitmap.compress(Bitmap.CompressFormat.PNG, 100,
	baos);
36	byte[]b = baos.toByteArray();
37	encodedImage = Base64.encodeToString(b,
	Base64.DEFAULT); }
38	//creating request parameters
39	HashMap<String, String> params = new HashMap<>();
40	params.put("idcivitas", inUsername);
41	params.put("title", inTitle);
42	params.put("idasset", inIDAsset);
43	params.put("description", inDescription);
44	params.put("suggestion", inSuggestion);
45	params.put("reportimage", encodedImage);
46	params.put("mobileapps", "true");
47	//kirim laporan
48	return
	requestHandler.sendPostRequest(URLs.URL_SENDREREPORT, params);}
49	@Override
	protected void onPostExecute(String s){
50	super.onPostExecute(s);
51	progressBar.setVisibility(View.GONE);
52	try {
53	JSONObject obj = new JSONObject(s);
54	//jika tidak ada error
55	if (!obj.getBoolean("error")) {
56	Toast.makeText(activity,
57	obj.getString("message"), Toast.LENGTH_SHORT).show();
	//kembali ke halaman laporan lini masa
58	activity.finish();
59	startActivity(new Intent(activity,
	MainActivity.class));

No	AddReportFragment.java
60	} else {
61	Toast.makeText(activity, "Error Occured", Toast.LENGTH_SHORT).show();}
62	} catch (JSONException e) {
63	e.printStackTrace();}}
64	SendReport sr = new SendReport();
65	sr.execute();}

Kode program baris 3-18 pada Tabel 6.1 merupakan implementasi dari skenario alternatif *use case* UCP-03, yaitu apabila pelapor belum mengisi seluruh masukan *form* untuk membuat laporan baru maka sistem memberikan pesan peringatan bahwa *form* laporan baru belum diisikan semua. Apabila pelapor melakukan *submit* ketika seluruh masukan *form* telah diisikan maka selanjutnya sistem menyimpan masukan judul laporan, ID aset, deskripsi laporan, saran perbaikan, dan foto laporan ke dalam variabel *string*. Kemudian menyimpan data dari ID civitas ke dalam variable *string* yang didapatkan dari *getter* setelah instansiasi objek User. Kemudian selanjutnya mengirimkan masukan tersebut ke web service melalui metode POST.

Tabel 6.2 Implementasi Fungsi Kamera dan Pindai QR Code

No	AddReportFragment.java
1	@Override
2	public void onActivityResult(int requestCode, int resultCode, Intent data) {
3	super.onActivityResult(requestCode, resultCode, data);
4	if (buttonLastClicked == 0){
5	Bitmap bitmap = (Bitmap) data.getExtras().get("data");
6	imageViewPhoto.setImageBitmap(bitmap);
7	ByteArrayOutputStream baos = new ByteArrayOutputStream();
8	bitmap.compress(Bitmap.CompressFormat.JPEG, 100, baos);
9	byte[]b = baos.toByteArray();
10	encodedImage = Base64.encodeToString(b, Base64.DEFAULT);
11	//Toast.makeText(this.getActivity(), encodedImage, Toast.LENGTH_SHORT).show();
12	stepBean4 = new StepBean("Foto",1);
13	stepBean5 = new StepBean("Selesai",1);
14	setStepView();}
15	else if (buttonLastClicked == 1) {
16	IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
17	if (result != null){
18	if (result.getContents() == null){
19	

No	AddReportFragment.java
20	<code>Toast.makeText(this.getActivity(), "You cancelled the scanning", Toast.LENGTH_SHORT).show();</code>
21	<code>else {</code>
22	<code>editTextIDAsset.setText(result.getContents().split("-")[0]);</code>
23	<code>editTextAssetName.setText(result.getContents().split("-")[1]);</code>
24	<code>editTextAssetType.setText(result.getContents().split("-")[2]);</code>
25	<code>editTextAssetLocation.setText(result.getContents().split("-")[3]);</code>
26	<code>else {</code>
27	<code>Toast.makeText(this.getActivity(), "Failed", Toast.LENGTH_SHORT).show();</code>
28	<code>//textViewQRResult.setText("Failed");</code>
	<code>//super.onActivityResult(requestCode, resultCode, data);}}}</code>

Pada skenario alternatif *use case* UCP-03 juga disebutkan bahwa pelapor dapat memilih fungsi gunakan kamera untuk mengambil foto laporan dan memilih fungsi pindai *QR Code* untuk memindai kode barang. Tabel 6.2 merupakan implementasi dari skenario alternatif untuk fungsi gunakan kamera dan pindai *QR Code* tersebut. Kode program pada baris 4 menjelaskan bahwa apabila tombol yang dipilih merupakan tombol kamera, setelah kamera dari perangkat telah mengambil gambar maka fungsi tersebut akan menyimpan hasil foto dari kamera. Sedangkan pada baris 15 menjelaskan bahwa apabila tombol yang dipilih merupakan tombol pindai, setelah kamera dari perangkat telah memindai kode QR, maka fungsi tersebut akan menyimpan hasil pindai ke dalam *edit text* yang tersedia.

6.2.1.2 Implementasi Melihat Laporan Lini Masa

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian perancangan sequence diagram melihat laporan lini masa dan perancangan algoritme melihat laporan yang telah dijelaskan pada Tabel 5.2 Perancangan Algoritme Melihat Laporan ke dalam wujud kode program pada kelas *TimelineFragment.java*. Implementasi kode program dari melihat laporan dijelaskan pada Tabel 6.3 berikut.

Tabel 6.3 Implementasi Melihat Laporan Lini Masa

No	TimelineFragment.java
1	<code>private void loadTimelineData() {</code>
2	<code>class RequestTimelineData extends AsyncTask<Void, Void, String></code>
3	<code>{</code>
4	<code>@Override</code>
5	<code>protected void onPreExecute(){</code>
6	<code>super.onPreExecute();</code>
7	<code>progressBar.setVisibility(View.VISIBLE);</code>
8	<code>}</code>
	<code>@Override</code>

No	TimelineFragment.java
9	protected String doInBackground(Void... voids) {
10	RequestHandler requestHandler = new RequestHandler();
11	return
	requestHandler.sendGetRequest (URLs.URL_REQUESTTIMELINE);
12	}
13	@Override
14	protected void onPostExecute(String s){
15	super.onPostExecute(s);
16	progressBar.setVisibility(View.GONE);
17	try {
18	JSONObject obj = new JSONObject(s);
19	//jika tidak terdapat error
20	if (!obj.getBoolean("error")) {
21	Toast.makeText(getActivity(),
22	obj.getString("message"), Toast.LENGTH_SHORT).show();
23	//get seluruh laporan dari balasan web service
	JSONArray timelineJSON =
24	obj.getJSONArray("timelinereports");
25	for (int i = 0; i<timelineJSON.length(); i++){
	JSONObject reportJSON =
26	timelineJSON.getJSONObject(i);
	byte[] decodedBytes =
27	Base64.decode(reportJSON.getString("REPORTIMAGE"),Base64.DEFAULT);
	Bitmap decodedBitmap =
28	BitmapFactory.decodeByteArray(decodedBytes, 0,
	decodedBytes.length);
29	Report report = new Report (
30	reportJSON.getString("IDREPORT"),
31	reportJSON.getString("REPORTERNAME"),
32	reportJSON.getString("TECHNICIANNAME"),
33	reportJSON.getString("ASSETNAME"),
34	reportJSON.getString("TIMESTART"),
35	reportJSON.getString("TIMEUPDATE"),
36	reportJSON.getString("TITLE"),
37	reportJSON.getString("DESCRIPTION"),
38	reportJSON.getString("SUGGESTION"),
39	reportJSON.getString("REPORTSTATUS"),
40	reportJSON.getString("SURVEYRESULT"),
41	reportJSON.getString("ENDRESULT"),
42	decodedBitmap);
43	

No	TimelineFragment.java
44	reportList.add(report);}}
45	else {
46	Toast.makeText(getActivity(), "Error Occured", Toast.LENGTH_SHORT).show();}
47	} catch (JSONException e) {
48	e.printStackTrace();}}
49	RequestTimelineData rTD = new RequestTimelineData();
50	rTD.execute();
	adapter = new TimelineAdapter(reportList, this.getContext());
	recyclerView.setAdapter(adapter);}

Kode program pada Tabel 6.3 merupakan implementasi dari *use case* UCP-04, yaitu untuk melihat laporan lini masa. Fungsi `loadTimelineData()` pada kelas `TimelineFragment.java` akan dipanggil ketika pelapor mengakses fungsi tampilkan daftar lini masa seluruh laporan pada menu utama aplikasi sesuai dengan kondisi awal dari *use case* UCP-04. Kode program pada baris 10-11 pada Tabel 6.3 merupakan baris yang berfungsi untuk melakukan permintaan data laporan lini masa ke *web service* melalui metode GET. Kemudian apabila balasan dari *web service* tidak mengandung error, maka sistem menyimpan data-data laporan timelineJSON dalam format JSON ke dalam objek `Report` dan selanjutnya objek-objek laporan tersebut ditampilkan dalam *RecyclerView*.

6.2.2 Implementasi Website Operator

Pada bagian ini merupakan implementasi dari *website* untuk *operator*. Implementasi program yang akan dijelaskan pada bagian ini terdiri dari penjelasan mengenai implementasi untuk menerima laporan, implementasi untuk menindak lanjuti laporan dan implementasi untuk mendisposisi teknisi sesuai dengan kebutuhan yang dijelaskan pada perancangan *sequence diagram operator* dan perancangan algoritme *operator*.

6.2.2.1 Implementasi Menerima Laporan

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian perancangan *sequence diagram* menerima laporan dan perancangan algoritme menerima laporan yang telah dijelaskan pada Tabel 5.3 Perancangan Algoritme Menerima Laporan ke dalam wujud kode program pada beberapa kelas dan potongan kode program. Implementasi untuk mendapatkan dan menampilkan notifikasi laporan dijelaskan pada Tabel 6.4 berikut.

Tabel 6.4 Implementasi Javascript Notifikasi Menerima Laporan

No	dashboard_footer.php
1	\$(document).ready(getReportNotification());
2	function getReportNotification() {
3	\$.get("<?php

No	dashboard_footer.php
	<pre> echo site_url("ReportController/reportNotificationRequest/");?>", function(data){ 4 var parsedata = JSON.parse(data); 5 for (var i = 0; i < parsedata.length; i++) { 6 document.getElementById("notifdetail"+i).href = "<?php echo site_url('ReportController/reportDetail/');?>" + parsedata[i].IDREPORT; 7 document.getElementById("notifimage"+i).src = 'data:assetimage.jpg;base64,' + parsedata[i].REPORTIMAGE; 8 document.getElementById("notiftitle"+i).innerHTML = parsedata[i].TITLE; 9 if (Math.floor(parsedata[i].SELISIHWAKTU/60) >= 24) { 10 document.getElementById("notiftime"+i).innerHTML = Math.floor(parsedata[i].SELISIHWAKTU/60/24)+' day ago'; 11 } else if (parsedata[i].SELISIHWAKTU >= 60) { 12 document.getElementById("notiftime"+i).innerHTML = Math.floor(parsedata[i].SELISIHWAKTU/60)+' hour ago'; 13 } else if (parsedata[i].SELISIHWAKTU < 60) { 14 document.getElementById("notiftime"+i).innerHTML = parsedata[i].SELISIHWAKTU+' minutes ago'; 15 }} }); setTimeout(getReportNotification,5000);}</pre>

Kode program pada Tabel 6.4 merupakan implementasi dari *use case* UCO-03, yaitu untuk melihat laporan bagian skenario alternatif memilih fungsi lihat notifikasi masuk. Fungsi `getReportNotification()` pada `dashboard_footer.php` akan dipanggil ketika halaman dari *website operator* telah selesai dimuat dan akan dipanggil kembali setiap 5 detik sekali. Kode program pada baris 3 pada Tabel 6.4 merupakan baris yang berfungsi untuk melakukan permintaan data laporan notifikasi kepada fungsi `reportNotificationRequest()` pada kelas *ReportController*. Setelah data balasan dari fungsi tersebut telah didapatkan, selanjutnya data laporan ditampilkan sebagai notifikasi pada halaman yang telah dimuat. Sedangkan implementasi dari fungsi `reportNotificationRequest()` pada kelas *ReportController* dijelaskan pada Tabel 6.5 berikut.

Tabel 6.5 Implementasi Controller ReportController Menerima Laporan

No	ReportController.php
1	<code>public function reportNotificationRequest(){</code>
2	<code> \$notificationreport = \$this->Report->getReportNotification(5);</code>
3	<code> \$notificationreportjson = json_encode(\$notificationreport);</code>
	<code> print_r(\$notificationreportjson);}</code>

Kode program pada Tabel 6.5 merupakan implementasi dari fungsi `reportNotificationRequest()` di kelas *ReportController* yang dipanggil pada fungsi `getReportNotification()` pada `dashboard_footer.php`. Fungsi ini berfungsi untuk mendapatkan data laporan pada kelas *model Report* lalu menampilkan data laporan tersebut dalam format JSON.

Tabel 6.6 Implementasi Model Report Menerima Laporan

No	Report.php
1	public function getReportNotification(\$limit){
2	\$value = \$this->db->query("SELECT IDREPORT, FLOOR(TIME_TO_SEC(TIMEDIFF(CURRENT_TIMESTAMP, TIMESTART))/60) AS SELISIHWAKTU, TITLE, REPORTIMAGE FROM REPORT ORDER BY IDREPORT DESC LIMIT ".\$limit."");
3	\$result = \$value->result_array();
4	return \$result;}

Kode program pada Tabel 6.6 merupakan implementasi dari fungsi `getReportNotification()` di kelas *model* Report yang dipanggil di fungsi `reportNotificationRequest()` pada kelas *ReportController*. Fungsi ini berfungsi untuk mendapatkan data laporan untuk notifikasi dari tabel *Report* dan mengembalikannya kepada fungsi `reportNotificationRequest()`.

6.2.2.2 Implementasi Menindaklanjuti Laporan

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian perancangan *sequence diagram* menindaklanjuti laporan dan perancangan algoritme menindaklanjuti laporan yang telah dijelaskan pada Tabel 5.4 Perancangan Algoritme Menindaklanjuti Laporan ke dalam wujud kode program pada beberapa kelas dan potongan kode program. Implementasi kode program untuk menindaklanjuti laporan dijelaskan pada Tabel 6.7 berikut.

Tabel 6.7 Implementasi Controller ReportController Menindaklanjuti Laporan

No	ReportController.php
1	public function reportDetailHandler(){
2	// jika tombol record dipilih
3	if (!empty(\$this->input->post('record')) && \$this->input->post('record') == 'record') {
4	redirect('ReportController/reportDetailRecord/'.\$this->input->post('idreport'));}
5	// jika tombol submit dipilih
6	else if (!empty(\$this->input->post('submit')) && \$this->input->post('submit') == 'submit') {
7	\$idreport = \$this->input->post('idreport');
8	\$idtechnician = \$this->input->post('idtechnician');
9	\$idasset = \$this->input->post('idasset');
10	\$description = \$this->input->post('description');
11	\$reportstatus = \$this->input->post('reportstatus');
12	\$surveyresult = \$this->input->post('surveyresult');
13	\$endresult = \$this->input->post('endresult');
14	// laporan sedang perbaikan lanjut disposisi teknisi
15	if (\$reportstatus == '7') {

No	ReportController.php
16	<code>\$endresult = '(Sedang menunggu hasil perbaikan)';</code>
17	<code>\$this->Report-</code> <code>>updateReportDetail(\$idreport,\$idasset,\$description,\$reportstatus,\$</code> <code>surveyresult,\$endresult);</code>
18	<code>\$this->ReportRecord->insertReportRecord(\$idreport,</code> <code>\$reportstatus, \$_SESSION['esid']);</code>
19	<code>redirect('ReportController/reportDispositionPage/'.\$this-></code> <code>input->post('idreport'));</code>
20	<code>// laporan selesai teknisi selesai kerja</code>
21	<code>else if (\$reportstatus == '8') {</code>
22	<code>\$this->Report-</code> <code>>updateReportDetail(\$idreport,\$idasset,\$description,\$reportstatus,\$</code> <code>surveyresult,\$endresult);</code>
23	<code>\$this->Technician->updateTechnicianStatus(\$idtechnician,</code> <code>1);</code>
24	<code>\$this->ReportRecord->insertReportRecord(\$idreport,</code> <code>\$reportstatus, \$_SESSION['esid']);</code>
25	<code>redirect('ReportController/reportDetail/'.\$this->input-</code> <code>>post('idreport'));</code>
26	<code>// update status selain sedang perbaikan dan selesai serta</code> <code>update data laporan biasa</code>
27	<code>else {</code>
28	<code>\$this->Report-</code> <code>>updateReportDetail(\$idreport,\$idasset,\$description,\$reportstatus,\$</code> <code>surveyresult,\$endresult);</code>
29	<code>\$this->ReportRecord->insertReportRecord(\$idreport,</code> <code>\$reportstatus, \$_SESSION['esid']);</code>
30	<code>redirect('ReportController/reportDetail/'.\$this->input-</code> <code>>post('idreport'));</code>

Kode program pada Tabel 6.7 merupakan implementasi dari *use case* UCO-04, yaitu menyediakan fungsi untuk mengubah data laporan dan statusnya serta menyimpan keterangan dari perubahan status pelaporannya. Kode program fungsi `reportDetailHandler()` akan dipanggil ketika *operator* telah men-*submit form* pada halaman rincian laporan. Kode program pada baris 7-13 berfungsi untuk menyimpan masukan pembaharuan data laporan dari operator ke dalam variabel. Kemudian dilanjutkan ke baris 15-30 untuk menentukan alur penyimpanan perubahan data berdasarkan status laporan terbaru yang merupakan implementasi dari skenario utama *use case* UCO-04. Apabila status laporan adalah sedang diperbaiki, maka perubahan data laporan akan disimpan dan akan dilanjutkan untuk proses disposisi teknisi. Apabila status laporan adalah selesai, maka perubahan data laporan akan disimpan dan status ketersediaan teknisi kembali menjadi tersedia. Sedangkan apabila status laporan selain dari sedang dikerjakan dan selesai maka akan menyimpan perubahan data laporan saja.

Tabel 6.8 Implementasi Model Report Menindaklanjuti Laporan

No	Report.php
1	public function updateReportDetail
2	(\$idreport,\$idasset,\$description,\$reportstatus,\$surveyresult,\$endresult){
3	<pre> \$this->db->query("UPDATE report SET IDASSET = '". \$idasset."', TIMEUPDATE = CURRENT_TIMESTAMP, DESCRIPTION = '". \$description."', IDREPORTSTATUS = '". \$reportstatus."', SURVEYRESULT = '". \$surveyresult."', ENDRESULT = '". \$endresult.'" WHERE IDREPORT = '". \$idreport.'" ");}</pre>

Penyimpanan perubahan data laporan dilakukan dengan cara memanggil fungsi `updateReportDetail()` pada kelas *model Report*. Implementasi dari fungsi `updateReportDetail()` pada kelas *model Report* dijelaskan pada Tabel 6.8. Fungsi `updateReportDetail()` menjalankan *query* untuk melakukan *update* pada tabel *report* untuk menyimpan perubahan data laporan yang sedang ditindaklanjuti.

6.2.2.3 Implementasi Mendisposisi Teknisi

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian perancangan *sequence diagram* mendisposisi teknisi ke dalam wujud kode program pada beberapa kelas dan potongan kode program. Implementasi kode program untuk mendisposisi teknisi dijelaskan pada Tabel 6.9 berikut.

Tabel 6.9 Implementasi Controller ReportController Mendisposisi Teknisi

No	ReportController.php
1	public function reportDispositionHandler(\$idreportandidtechnician){
2	\$params = explode('-', \$idreportandidtechnician);
3	\$idreport = \$params[0];
4	\$idtechnician = \$params[1];
5	\$this->Report->updateReportTechnician(\$idreport, \$idtechnician);
6	\$this->Technician->updateTechnicianStatus(\$idtechnician,2);
7	\$this->reportDetail(\$idreport);}

Kode program pada Tabel 6.9 merupakan implementasi dari *use case* UCO-10, yaitu untuk melakukan disposisi teknisi. Kode program fungsi `reportDispositionHandler()` akan dipanggil ketika *operator* telah men-*submit form* pada halaman disposisi teknisi. Kode program pada baris 2-4 berfungsi untuk menyimpan data ID laporan yang ditindak lanjuti dan ID teknisi yang diberikan disposisi ke dalam variabel. Selanjutnya dilanjutkan dengan menyimpan data teknisi yang melakukan perbaikan terhadap laporan yang ditindak lanjuti seperti yang dijelaskan pada Tabel 6.10 dan merubah status teknisi tersebut menjadi sedang bekerja seperti yang dijelaskan pada Tabel 6.11.

Tabel 6.10 Implementasi Model Report Menindaklanjuti Laporan

No	Report.php
1	public function updateReportTechnician(\$idreport, \$idtechnician){
2	\$this->db->query("UPDATE REPORT SET IDTECHNICIAN = '".\$idtechnician."', IDREPORTSTATUS = 7 WHERE IDREPORT = '".\$idreport."'");}

Kode program pada Tabel 6.10 merupakan implementasi dari fungsi `updateReportTechnician()` pada kelas *model Report* yang berisi *query* untuk melakukan *update* terhadap tabel *report* mengenai penyimpanan data teknisi yang melakukan perbaikan terhadap laporan yang sedang ditindak lanjuti.

Tabel 6.11 Implementasi Model Technician Menindaklanjuti Laporan

No	Technician.php
1	public function updateTechnicianStatus(\$idtechnician, \$idstatus){
2	\$this->db->query("UPDATE technician SET IDTECHNICIANSTATUS = '".\$idstatus."' WHERE IDTECHNICIAN = '".\$idtechnician."'");}

Sedangkan kode program pada Tabel 6.11 merupakan implementasi dari fungsi `updateTechnicianStatus()` pada kelas *model Technician* yang berisi *query* untuk melakukan *update* terhadap tabel *technician* mengenai perubahan status teknisi yang diberikan disposisi menjadi sedang bekerja.

6.2.3 Implementasi Web Service

Pada bagian ini merupakan implementasi dari *web service* yang disediakan untuk menghubungkan aplikasi *mobile* pelapor dengan basis data dari Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Implementasi *web service* yang akan dijelaskan pada bagian ini terdiri dari penjelasan mengenai implementasi fungsi *web service* untuk mendaftarkan akun, menambahkan laporan dan mendapatkan laporan yang terbagi pada beberapa kelas *controller*.

6.2.3.1 Implementasi Web Service Mendaftarkan Akun

Bagian ini merupakan bagian implementasi dari *web service* untuk fungsi mendaftarkan akun dari aplikasi *mobile* pelapor ke dalam basis data Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Implementasi *web service* ini merupakan bagian dari implementasi dari *use case* UCP-01 yaitu sistem dapat menyimpan data akun pelapor agar *guest* bisa mendapatkan hak akses terhadap sistem. Implementasi kode program *web service* untuk fungsi mendaftarkan akun dijelaskan pada Tabel 6.12 berikut.

Tabel 6.12 Implementasi Controller ReportController Mendaftarkan Akun

No	CivitasController.php
1	public function doRegister(){
2	\$IDCivitas = \$this->input->post('IDCivitas');
3	\$Fullname = \$this->input->post('Fullname');
4	\$Password = \$this->input->post('Password');

No	CivitasController.php
5	<code>\$Email = \$this->input->post('Email');</code>
6	<code>\$Gender = \$this->input->post('Gender');</code>
7	<code>\$Phoneno = \$this->input->post('Phoneno');</code>
8	<code>\$Civitastype = \$this->input->post('Civitastype');</code>
9	<code>\$CivitasImage = \$this->input->post('IDImage');</code>
10	<code>\$this->Civitas->setCivitas(\$IDCivitas, \$Fullname, \$Password, \$Email, \$Gender, \$Phoneno, \$CivitasImage);</code>
11	<code>\$response = array();</code>
12	<code>if (\$Civitastype == 'Student') {</code>
13	<code> \$NIM = \$this->input->post('NIM');</code>
14	<code> \$IDMajor = \$this->input->post('IDMajor');</code>
15	<code> \$IDProgram = \$this->input->post('IDProgram');</code>
16	<code> \$Year = \$this->input->post('Year');</code>
17	<code> \$this->Student->setStudent(\$NIM, \$IDCivitas, \$IDMajor, \$IDProgram, \$Year);</code>
18	<code> \$response['error'] = false;</code>
19	<code> \$response['message'] = 'Student succesfully registered';</code>
20	<code>} else if (\$Civitastype == 'Lecturer') {</code>
21	<code> \$NIP = \$this->input->post('NIP');</code>
22	<code> \$IDMajor = \$this->input->post('IDMajor');</code>
23	<code> \$this->Lecturer->setLecturer(\$NIP, \$IDCivitas, \$IDMajor);</code>
24	<code> \$response['error'] = false;</code>
25	<code> \$response['message'] = 'Lecturer succesfully registered';</code>
26	<code>} else if (\$Civitastype == 'Staff') {</code>
27	<code> \$NIP = \$this->input->post('NIP');</code>
28	<code> \$Position = \$this->input->post('Position');</code>
29	<code> \$this->Staff->setStaff(\$NIP, \$IDCivitas, \$Position);</code>
30	<code> \$response['error'] = false;</code>
31	<code> \$response['message'] = 'Staff succesfully registered';</code>
32	<code>\$response = json_encode(\$response, JSON_PRETTY_PRINT);</code>
33	<code>print_r(\$response);}</code>

Kode program pada Tabel 6.12 merupakan fungsi yang terdapat pada kelas *controller* CivitasController.php dan dapat diakses dari aplikasi *mobile* pelapor dengan mengirimkan parameter masukan melalui metode POST. Kode program baris 2-9 menjelaskan parameter masukan yang perlu dikirimkan dari aplikasi *mobile* pelapor untuk mendaftarkan akun dan dilanjutkan untuk menyimpan data tersebut ke dalam basis data pada baris ke 10. Kemudian baris 12, 20 dan 26 menjelaskan mengenai jenis civitas dari pelapor yang sedang didaftarkan, yang kemudian dilanjutkan dengan menyimpan data pelapor ke dalam basis data berdasarkan jenis civitasnya.

6.2.3.2 Implementasi Web Service Menambahkan Laporan

Bagian ini merupakan bagian implementasi dari *web service* untuk fungsi menambahkan laporan dari aplikasi *mobile* pelapor ke dalam basis data Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Implementasi *web service* ini merupakan bagian dari implementasi dari *use case* UCP-03 yaitu sistem dapat mengirimkan laporan keluhan ke *operator*. Implementasi kode program *web service* untuk fungsi menambahkan laporan dijelaskan pada Tabel 6.13 berikut.

Tabel 6.13 Implementasi Controller ReportController Menambahkan Laporan

No	ReportController.php
1	<code>public function reportAddHandler() {</code>
2	<code> \$title = \$this->input->post('title');</code>
3	<code> \$idasset = \$this->input->post('idasset');</code>
4	<code> \$description = \$this->input->post('description');</code>
5	<code> \$suggestion = \$this->input->post('suggestion');</code>
6	<code> if (\$this->input->post('webadmin')) {</code>
7	<code> \$sidcivitas = \$_SESSION['esidcivitas'];</code>
8	<code> if (\$_FILES['reportimage']['size'] != 0) {</code>
9	<code> \$reportimagetype =</code>
10	<code> pathinfo(\$_FILES['reportimage']['tmp_name'], PATHINFO_EXTENSION);</code>
11	<code> \$reportimagedata =</code>
12	<code> file_get_contents(\$_FILES['reportimage']['tmp_name']);</code>
13	<code> \$reportimagebase64 = base64_encode(\$reportimagedata);}</code>
14	<code> else {</code>
15	<code> \$this->load->model('Model_NoImage');</code>
16	<code> \$reportimagebase64 = \$this->Model_NoImage->getNotAvailableImageBase64();}</code>
17	<code> \$this->Report->insertNewReport(\$sidcivitas, \$title, \$idasset,</code>
18	<code> \$description, \$suggestion, \$reportimagebase64);</code>
19	<code> \$this->reportPage();}</code>
20	<code> else if (\$this->input->post('mobileapps')) {</code>
21	<code> \$sidcivitas = \$this->input->post('idcivitas');</code>
22	<code> \$reportimagebase64 = \$this->input->post('reportimage');</code>
23	<code> \$this->Report->insertNewReport(\$sidcivitas, \$title, \$idasset,</code>
24	<code> \$description, \$suggestion, \$reportimagebase64);</code>
25	<code> \$report = array();</code>
26	<code> \$report['error'] = false;</code>
27	<code> \$report['message'] = 'Report '.\$title.' submitted</code>
28	<code> succesfully';</code>
29	<code> \$report = json_encode(\$report);</code>
30	<code> print_r(\$report);}}</code>

Kode program pada Tabel 6.13 merupakan fungsi yang terdapat pada kelas *controller ReportController* dan dapat diakses dari aplikasi *mobile* pelapor dengan mengirimkan parameter masukan melalui metode POST. Kode program baris 2-5

menjelaskan parameter masukan yang perlu dikirimkan dari aplikasi *mobile* pelapor untuk menambahkan laporan. Kemudian dilanjutkan dengan seleksi kondisi pada baris 6 dan 17, apabila data dikirimkan dari aplikasi *mobile* maka alur dilanjutkan pada baris 18-25 yang berfungsi untuk menyimpan data laporan ke dalam basis data dan mengirimkan pesan balasan berupa pesan sukses.

6.2.3.3 Implementasi Web Service Mendapatkan Laporan

Bagian ini merupakan bagian implementasi dari *web service* untuk fungsi mendapatkan laporan dari basis data Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana untuk aplikasi *mobile* pelapor. Implementasi *web service* yang akan dijelaskan pada bagian ini diwakilkan dengan fungsi untuk mendapatkan laporan riwayat yang merupakan bagian dari implementasi dari *use case* UCP-06 yaitu untuk menampilkan keluhan yang pernah dilaporkan oleh pelapor yang telah diselesaikan atau telah ditolak. Implementasi kode program *web service* untuk fungsi mendapatkan laporan riwayat dijelaskan pada Tabel 6.14 berikut.

Tabel 6.14 Implementasi Controller ReportController Mendapatkan Laporan Riwayat

No	ReportController.php
1	public function getReportHistory(){
2	\$method = \$_SERVER['REQUEST_METHOD'];
3	\$idcivitas = \$this->input->post('idcivitas');
4	\$response = \$this->Report->getReportHistory(\$idcivitas);
5	\$response = json_encode(\$response, JSON_PRETTY_PRINT);
6	print_r(\$response);}

Kode program pada Tabel 6.14 merupakan fungsi yang terdapat pada kelas *controller ReportController* dan dapat diakses dari aplikasi *mobile* pelapor dengan mengirimkan parameter masukan melalui metode POST. Kode program baris 3 menjelaskan parameter masukan yang perlu dikirimkan dari aplikasi *mobile* pelapor untuk mendapatkan laporan riwayat. Kemudian dilanjutkan dengan memanggil fungsi pada *model Report* untuk mendapatkan data laporan riwayat. Implementasi fungsi dari kelas *model Report* dijelaskan pada Tabel 6.15 berikut.

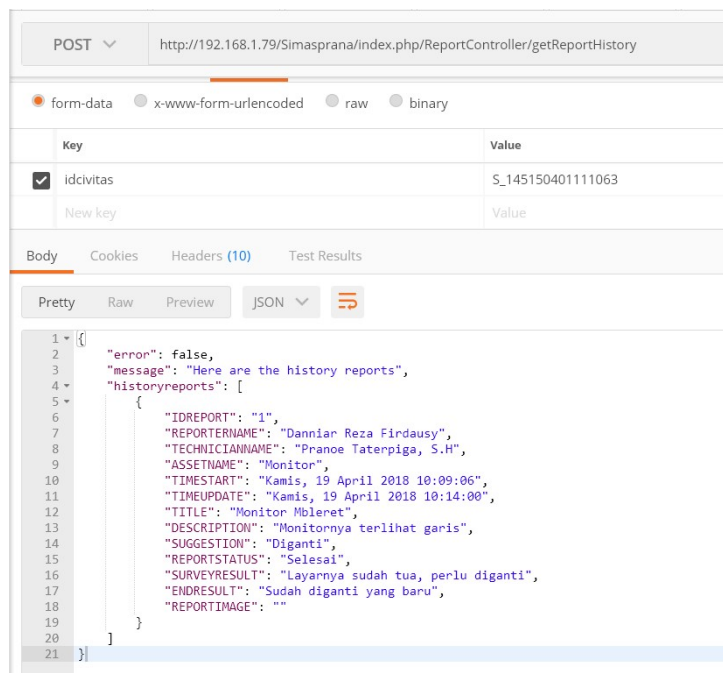
Tabel 6.15 Implementasi Model Report Mendapatkan Laporan Riwayat

No	Report.php
1	public function getReportHistory(\$idcivitas){
2	\$value = \$this->db->query("SELECT R.IDREPORT, CP.FULLNAME AS REPORTERNAME, C.FULLNAME AS TECHNICIANNAME, A.NAME AS ASSETNAME, R.TIMESTART, R.TIMEUPDATE, R.TITLE, R.DESCRPTION, R.SUGGESTION, RS.REPORTSTATUS, R.REPORTIMAGE, R.SURVEYRESULT AS SURVEYRESULT, R.ENDRESULT AS ENDRESULT FROM CIVITAS C JOIN TECHNICIAN T ON C.IDCIVITAS = T.IDCIVITAS RIGHT OUTER JOIN REPORT R ON T.IDTECHNICIAN = R.IDTECHNICIAN JOIN CIVITAS CP ON CP.IDCIVITAS = R.IDCIVITAS JOIN ASSET A ON R.IDASSET = A.IDASSET JOIN REPORTSTATUS RS ON r.IDREPORTSTATUS = rs.IDREPORTSTATUS WHERE R.IDCIVITAS =

No	Report.php
3	'".\$idcivitas."' AND R.IDREPORTSTATUS IN (8,9) ORDER BY R.TIMESTART DESC");
4	\$historyreports = array();
5	if (\$value->num_rows()<1) {
6	\$response['error'] = false;
7	\$response['message'] = 'History reports not found';
8	\$response['historyreports'] = NULL;
9	} else {
10	\$result = \$value->result_array();
11	foreach (\$result as \$row) {
12	\$report = array();
13	if (\$row['TECHNICIANNAME'] == NULL) {
14	\$row['TECHNICIANNAME'] = '(Belum Tersedia)';
15	if (\$row['SURVEYRESULT'] == NULL) {
16	\$row['SURVEYRESULT'] = '(Belum Tersedia)';
17	if (\$row['ENDRESULT'] == NULL) {
18	\$row['ENDRESULT'] = '(Belum Tersedia)';
19	\$findday = array("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat");
20	\$replaceday = array("Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jum'at", "Sabtu");
21	\$findmonth = array("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Des");
22	\$replacemonth = array("Januari", "Februari", "Maret", "April", "Mei", "Juni", "Juli", "Agustus", "September", "Oktober", "November", "Desember");
23	\$datestart = strtotime(\$row['TIMESTART']);
24	\$datestart = date('D, d M Y h:i:s', \$datestart);
25	\$datestart = str_replace(\$findday, \$replaceday, \$datestart);
26	\$datestart = str_replace(\$findmonth, \$replacemonth, \$datestart);
27	if (\$row['TIMEUPDATE'] != NULL) {
28	\$dateupdate = strtotime(\$row['TIMEUPDATE']);
29	\$dateupdate = date('D, d M Y h:i:s', \$dateupdate);
30	\$dateupdate = str_replace(\$findday, \$replaceday, \$dateupdate);
31	\$dateupdate = str_replace(\$findmonth, \$replacemonth, \$dateupdate);
32	} else {
33	\$dateupdate = '(Belum tersedia)';
34	\$report['IDREPORT'] = \$row['IDREPORT'];

No	Report.php
35	<code>\$report['REPORTERNAME'] = \$row['REPORTERNAME'];</code>
36	<code>\$report['TECHNICIANNAME'] = \$row['TECHNICIANNAME'];</code>
37	<code>\$report['ASSETNAME'] = \$row['ASSETNAME'];</code>
38	<code>\$report['TIMESTART'] = \$datestart;</code>
39	<code>\$report['TIMEUPDATE'] = \$dateupdate;</code>
40	<code>\$report['TITLE'] = \$row['TITLE'];</code>
41	<code>\$report['DESCRIPTION'] = \$row['DESCRIPTION'];</code>
42	<code>\$report['SUGGESTION'] = \$row['SUGGESTION'];</code>
43	<code>\$report['REPORTSTATUS'] = \$row['REPORTSTATUS'];</code>
44	<code>\$report['SURVEYRESULT'] = \$row['SURVEYRESULT'];</code>
45	<code>\$report['ENDRESULT'] = \$row['ENDRESULT'];</code>
46	<code>\$report['REPORTIMAGE'] = \$row['REPORTIMAGE'];</code>
47	<code>array_push(\$historyreports, \$report);}</code>
48	<code>\$response['error'] = false;</code>
49	<code>\$response['message'] = 'Here are the history reports';</code>
50	<code>\$response['historyreports'] = \$historyreports;</code>
	<code>} return \$response; }</code>

Kode program pada Tabel 6.15 merupakan fungsi yang terdapat pada kelas *model Report* untuk mendapatkan data laporan riwayat berdasarkan ID Civitas dari basis data. Kode program pada baris 4 dan 8 merupakan seleksi kondisi untuk menentukan apabila *query* tidak berisi kembalian data maka mengembalikan pesan laporan riwayat tidak tersedia kepada aplikasi dan jika ada mengembalikan data-data laporan riwayat dalam format JSON kepada aplikasi.



Gambar 6.1 Format JSON Mendapatkan Laporan Riwayat

Pada Gambar 6.1 merupakan implementasi dari format data kembalian dalam JSON dari fungsi *web service* mendapatkan laporan riwayat. Bagian paling atas dari Gambar 6.1 merupakan alamat URL dari *web service* yang bisa diakses oleh aplikasi untuk mengakses *web service* mendapatkan laporan riwayat.

6.2.4 Implementasi Data Definition Language (DDL)

Bagian ini berisi implementasi dari Data Definition Language (DDL) tabel-tabel yang digunakan untuk menyimpan data pada basis data Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana. Bagian implementasi DDL ini merupakan tahap implementasi yang mengacu kepada perancangan yang telah dilakukan sebelumnya pada bagian Perancangan Skema Basis Data. Pada implementasi DDL dari tabel-tabel ini dijelaskan berupa *query* yang digunakan untuk membuat tabel-tabel pada basis data. Berikut pada Tabel 6.16 menjelaskan tentang implementasi DDL untuk tabel Asset yang digunakan untuk menyimpan data-data dari sarana prasarana.

Tabel 6.16 Implementasi DDL Tabel Asset

No	Asset
1	CREATE TABLE asset (
2	IDASSET varchar(10) NOT NULL,
3	NAME varchar(30) NOT NULL,
4	IDASSETTYPE tinyint(4) NOT NULL,
5	IDLOCATION int(11) NOT NULL,
6	QUANTITY int(11) NOT NULL,

No	Asset
7	IDASSETSTATUS int(11) NOT NULL,
8	ASSETIMAGE mediumtext NOT NULL,
9	PRIMARY KEY (IDASSET),
10	FOREIGN KEY (IDASSETSTATUS) REFERENCES assetstatus (IDASSETSTATUS),
11	FOREIGN KEY (IDASSETTYPE) REFERENCES assettype (IDASSETTYPE),
12	FOREIGN KEY (IDLOCATION) REFERENCES location (IDLOCATION)
)

Berikut pada Tabel 6.17 menjelaskan tentang implementasi DDL untuk tabel Assetstatus yang digunakan untuk menyimpan data status dari aset.

Tabel 6.17 Implementasi DDL Tabel Assetstatus

No	Assetstatus
1	CREATE TABLE assetstatus (
2	IDASSETSTATUS int(11) NOT NULL AUTO_INCREMENT,
3	ASSETSTATUS varchar(20) NOT NULL,
4	PRIMARY KEY (IDASSETSTATUS)
)

Berikut pada Tabel 6.18 menjelaskan tentang implementasi DDL untuk tabel Assettype yang digunakan untuk menyimpan data dari jenis-jenis aset.

Tabel 6.18 Implementasi DDL Tabel Assettype

No	Assettype
1	CREATE TABLE assettype (
2	IDASSETTYPE tinyint(11) NOT NULL AUTO_INCREMENT,
3	ASSETTYPE varchar(15) NOT NULL,
4	PRIMARY KEY (IDASSETTYPE)
)

Berikut pada Tabel 6.19 menjelaskan tentang implementasi DDL untuk tabel Civitas yang digunakan untuk menyimpan data-data dari civitas akademik FILKOM.

Tabel 6.19 Implementasi DDL Tabel Civitas

No	Civitas
1	CREATE TABLE civitas (
2	IDCIVITAS varchar(30) NOT NULL,
3	FULLNAME varchar(50) NOT NULL,
4	PASSWORD varchar(20) NOT NULL,
5	EMAIL varchar(50) NOT NULL,
6	GENDER char(6) NOT NULL,
7	PHONENO varchar(15) NOT NULL,
8	CIVITASIMAGE mediumtext NOT NULL,
9	PRIMARY KEY (IDCIVITAS)

No	Civitas
)

Berikut pada Tabel 6.20 menjelaskan tentang implementasi DDL untuk tabel Student yang digunakan untuk menyimpan data-data mahasiswa civitas akademik FILKOM.

Tabel 6.20 Implementasi DDL Tabel Student

No	Student
1	CREATE TABLE student (
2	NIM char(15) NOT NULL,
3	IDCIVITAS varchar(30) NOT NULL,
4	IDMAJOR varchar(3) NOT NULL,
5	IDPROGRAM varchar(3) NOT NULL,
6	YEAR smallint(6) NOT NULL,
7	PRIMARY KEY (NIM, IDCIVITAS),
8	FOREIGN KEY (IDCIVITAS) REFERENCES civitas (IDCIVITAS),
9	FOREIGN KEY (IDMAJOR) REFERENCES major (IDMajor),
10	FOREIGN KEY (IDPROGRAM) REFERENCES program (IDProgram)
)

Berikut pada Tabel 6.21 menjelaskan tentang implementasi DDL untuk tabel Lecturer yang digunakan untuk menyimpan data-data dari dosen civitas akademik FILKOM.

Tabel 6.21 Implementasi DDL Tabel Lecturer

No	Lecturer
1	CREATE TABLE lecturer (
2	NIP char(20) NOT NULL,
3	IDCIVITAS varchar(30) NOT NULL,
4	IDMAJOR varchar(3) NOT NULL,
5	PRIMARY KEY (NIP, IDCIVITAS),
6	FOREIGN KEY (IDCIVITAS) REFERENCES civitas (IDCIVITAS),
7	FOREIGN KEY (IDMAJOR) REFERENCES major (IDMajor)
)

Berikut pada Tabel 6.22 menjelaskan tentang implementasi DDL untuk tabel Staff yang digunakan untuk menyimpan data-data dari pegawai civitas akademik FILKOM.

Tabel 6.22 Implementasi DDL Tabel Staff

No	Staff
1	CREATE TABLE staff (
2	NIP char(20) NOT NULL,
3	IDCIVITAS varchar(30) NOT NULL,
4	POSITION varchar(30) NOT NULL,

No	Staff
5	PRIMARY KEY (NIP, IDCIVITAS),
6	FOREIGN KEY (IDCIVITAS) REFERENCES civitas (IDCIVITAS)
)

Berikut pada Tabel 6.23 menjelaskan tentang implementasi DDL untuk tabel EquipmentStaff yang digunakan untuk menyimpan data-data dari pegawai perlengkapan yang merupakan bagian dari pegawai civitas akademik FILKOM.

Tabel 6.23 Implementasi DDL Tabel EquipmentStaff

No	EquipmentStaff
1	CREATE TABLE equipmentstaff (
2	IDEQUIPMENTSTAFF varchar(15) NOT NULL,
3	NIP char(20) NOT NULL,
4	IDCIVITAS varchar(30) NOT NULL,
5	PRIMARY KEY (IDEQUIPMENTSTAFF, IDCIVITAS, NIP),
6	FOREIGN KEY (IDCIVITAS) REFERENCES civitas (IDCIVITAS),
7	FOREIGN KEY (NIP) REFERENCES staff (NIP)
)

Berikut pada Tabel 6.24 menjelaskan tentang implementasi DDL untuk tabel Technician yang digunakan untuk menyimpan data-data dari teknisi yang merupakan bagian dari pegawai civitas akademik FILKOM.

Tabel 6.24 Implementasi DDL Tabel Technician

No	Technician
1	CREATE TABLE technician (
2	IDTECHNICIAN varchar(15) NOT NULL,
3	IDCIVITAS varchar(30) NOT NULL,
4	NIP char(20) NOT NULL,
5	WORKPLACE varchar(20) NOT NULL,
6	IDTECHNICIANSTATUS int(11) NOT NULL,
7	PRIMARY KEY (IDTECHNICIAN, IDCIVITAS, NIP),
8	FOREIGN KEY (IDCIVITAS) REFERENCES civitas (IDCIVITAS),
9	FOREIGN KEY (IDTECHNICIANSTATUS) REFERENCES technicianstatus
10	(IDTECHNICIANSTATUS),
11	FOREIGN KEY (NIP) REFERENCES staff (NIP)
)

Berikut pada Tabel 6.25 menjelaskan tentang implementasi DDL untuk tabel TechnicianStatus yang digunakan untuk menyimpan data dari status ketersediaan teknisi.

Tabel 6.25 Implementasi DDL Tabel TechnicianStatus

No	Technician Status
1	CREATE TABLE technicianstatus (
2	IDTECHNICIANSTATUS int(11) NOT NULL AUTO_INCREMENT,
3	TECHNICIANSTATUS varchar(20) NOT NULL,
4	PRIMARY KEY (IDTECHNICIANSTATUS)
)

Berikut pada Tabel 6.26 menjelaskan tentang implementasi DDL untuk tabel Report yang digunakan untuk menyimpan data laporan keluhan sarana prasarana yang dilaporkan oleh civitas akademik FILKOM.

Tabel 6.26 Implementasi DDL Tabel Report

No	Report
1	CREATE TABLE report (
2	IDREPORT int(11) NOT NULL AUTO_INCREMENT,
3	IDCIVITAS varchar(30) NOT NULL,
4	IDREPORTSTATUS int(11) NOT NULL,
5	IDASSET varchar(10) NOT NULL,
6	IDTECHNICIAN varchar(15) DEFAULT NULL,
7	TIMESTART timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
8	TIMEUPDATE timestamp NULL DEFAULT '0000-00-00 00:00:00',
9	TITLE varchar(50) NOT NULL,
10	DESCRIPTION varchar(255) NOT NULL,
11	SUGGESTION varchar(255) NOT NULL,
12	REPORTIMAGE mediumtext,
13	SURVEYRESULT varchar(255) DEFAULT NULL,
14	ENDRESULT varchar(255) DEFAULT NULL,
15	PRIMARY KEY (IDREPORT),
16	FOREIGN KEY (IDASSET) REFERENCES asset (IDASSET),
17	FOREIGN KEY (IDCIVITAS) REFERENCES civitas (IDCIVITAS),
18	FOREIGN KEY (IDREPORTSTATUS) REFERENCES reportstatus (IDREPORTSTATUS),
19	FOREIGN KEY (IDTECHNICIAN) REFERENCES technician (IDTECHNICIAN)
)

Berikut pada Tabel 6.27 menjelaskan tentang implementasi DDL untuk tabel ReportStatus yang digunakan untuk menyimpan data dari status laporan keluhan sarana prasarana.

Tabel 6.27 Implementasi DDL Tabel ReportStatus

No	Report Status
1	CREATE TABLE reportstatus (
2	IDREPORTSTATUS int(11) NOT NULL AUTO_INCREMENT,
3	REPORTSTATUS varchar(30) NOT NULL,
4	PRIMARY KEY (IDREPORTSTATUS)
)

Berikut pada Tabel 6.28 menjelaskan tentang implementasi DDL untuk tabel ReportRecord yang digunakan untuk menyimpan data-data rekam pembaharuan data laporan keluhan sarana prasarana.

Tabel 6.28 Implementasi DDL Tabel ReportRecord

No	Report Record
1	CREATE TABLE reportrecord (
2	IDREPORTRECORD int(11) NOT NULL AUTO_INCREMENT,
3	IDREPORT int(11) NOT NULL,
4	IDREPORTSTATUS int(11) NOT NULL,
5	IDEQUIPMENTSTAFF varchar(15) NOT NULL,
6	TIMEUPDATE timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
7	PRIMARY KEY (IDREPORTRECORD),
8	FOREIGN KEY (IDEQUIPMENTSTAFF) REFERENCES equipmentstaff (IDEQUIPMENTSTAFF),
9	FOREIGN KEY (IDREPORT) REFERENCES report (IDREPORT),
10	FOREIGN KEY (IDREPORTSTATUS) REFERENCES reportstatus (IDREPORTSTATUS))

6.3 Implementasi Antarmuka

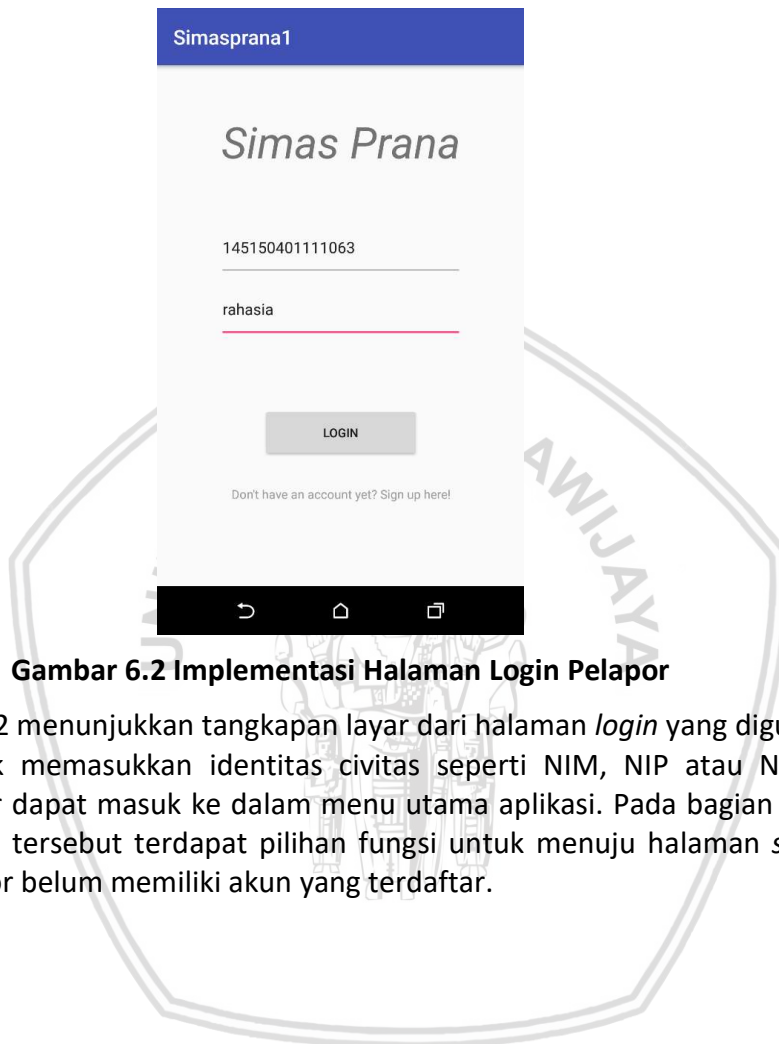
Pada bagian ini berisi implementasi dari antarmuka Sistem Informasi Manajemen yang dikembangkan. Pada implementasi antarmuka dari Sistem Informasi Manajemen ini terbagi menjadi dua bagian sesuai dengan aktor yang telah diidentifikasi, yaitu implementasi antarmuka aplikasi *mobile* pelapor dan implementasi antarmuka *website operator*.

6.3.1 Implementasi Antarmuka Aplikasi *Mobile* Pelapor

Pada bagian ini merupakan implementasi antarmuka dari aplikasi *mobile* pelapor. Implementasi antarmuka yang dijelaskan pada bagian ini terdiri dari penjelasan mengenai antarmuka untuk *login* dan *sign up* pelapor, halaman membuat laporan baru, halaman laporan serta halaman rincian laporan sesuai dengan kebutuhan yang dijelaskan pada perancangan antarmuka pengguna pelapor.

6.3.1.1 Antarmuka Halaman Login dan Sign Up Pelapor

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan halaman *login* dan *sign up* pelapor pada Gambar 5.15 ke dalam wujud tangkapan layar antarmuka. Implementasi antarmuka untuk halaman *login* dari pelapor ditunjukkan pada Gambar 6.2 berikut.



Gambar 6.2 Implementasi Halaman Login Pelapor

Gambar 6.2 menunjukkan tangkapan layar dari halaman *login* yang digunakan pelapor untuk memasukkan identitas civitas seperti NIM, NIP atau NIK dan *password* agar dapat masuk ke dalam menu utama aplikasi. Pada bagian bawah halaman *login* tersebut terdapat pilihan fungsi untuk menuju halaman *sign up* apabila pelapor belum memiliki akun yang terdaftar.

Gambar 6.3 Implementasi Halaman Sign Up Pelapor

Gambar 6.3 menunjukkan tangkapan layar dari halaman *sign up* yang digunakan pelapor untuk mendaftarkan akun sebagai pelapor. Pada halaman ini pelapor dapat memilih jenis civitas, memasukkan nomor identitas, nama lengkap, *email*, password, jenis kelamin dan nomor telepon. Setelah itu pelapor diharuskan mengambil foto dari kartu identitas yang dimiliki lalu dilanjutkan ke halaman berikutnya untuk melengkapi informasi mengenai jenis civitasnya.

6.3.1.2 Antarmuka Halaman Laporan Baru

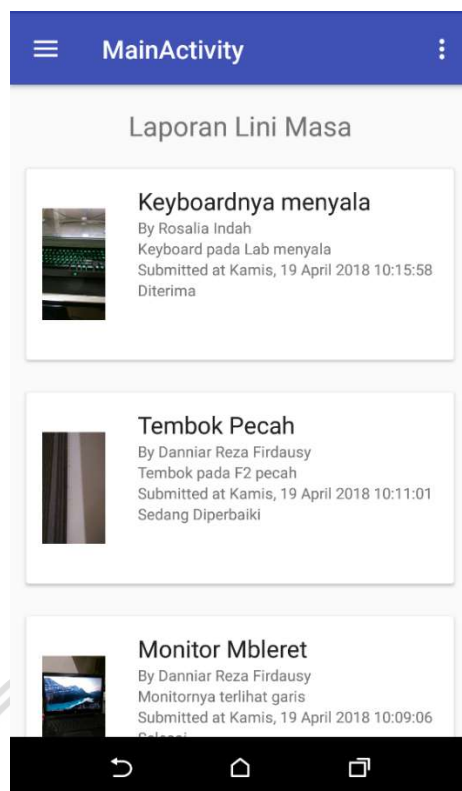
Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan halaman laporan baru pada Gambar 5.16 ke dalam wujud tangkapan layar antarmuka. Implementasi antarmuka untuk halaman laporan baru ditunjukkan pada Gambar 6.4 berikut.

Gambar 6.4 Implementasi Halaman Laporan Baru Inventaris

Gambar 6.4 menunjukkan tangkapan layar dari halaman laporan baru yang digunakan pelapor untuk membuat laporan baru dari keluhan barang inventaris atau barang aset yang terdaftar. Halaman laporan baru terdiri dari *form* yang berisi masukan judul laporan, deskripsi laporan, sarana tindakan, informasi dari aset, foto laporan dan beberapa tombol untuk fungsi kamera, pindai kode QR dan *submit*. Informasi mengenai aset yang dilaporkan hanya dapat diisi dengan menggunakan fungsi pindai kode QR dari aset tersebut yang kemudian setelah berhasil melakukan pemindaian informasi aset tersebut secara otomatis terisi. Foto laporan hanya dapat diisi dengan menggunakan fungsi kamera yang telah disediakan oleh perangkat secara default. Untuk membantu pelapor dalam mengisi *form*, pada awal halaman telah disediakan panduan langkah-langkah yang harus dilakukan dan telah selesai dilakukan oleh pelapor dalam mengisi *form* pelaporan.

6.3.1.3 Antarmuka Halaman Laporan Lini Masa, Sedang Berjalan dan Riwayat

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan halaman laporan lini masa, sedang berjalan dan riwayat pada Gambar 5.17 ke dalam wujud tangkapan layar antarmuka. Implementasi antarmuka untuk halaman laporan lini masa ditunjukkan pada Gambar 6.5 berikut.

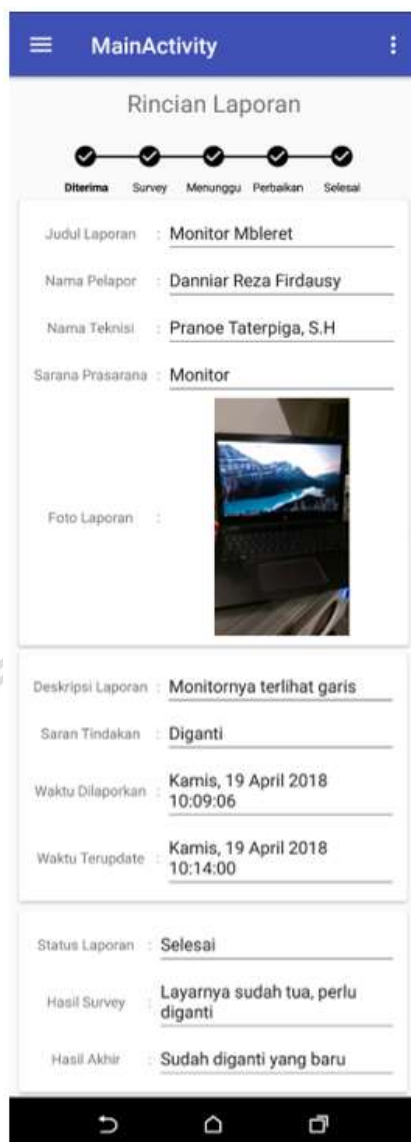


Gambar 6.5 Implementasi Halaman Laporan Lini Masa

Gambar 6.5 menunjukkan halaman yang menampilkan daftar laporan lini masa. Halaman laporan lini masa ini menampilkan seluruh keluhan yang telah dilaporkan oleh seluruh pelapor civitas akademik. Informasi dari laporan yang ditampilkan pada halaman ini antara lain adalah foto laporan, judul laporan, nama pelapor, deskripsi keluhan, waktu pelaporannya dan status pelaporannya. Implementasi antarmuka untuk halaman laporan yang sedang berjalan hanya menampilkan daftar laporan yang telah dilaporkan oleh pelapor yang sedang *login* dan status pelaporannya belum selesai atau belum ditolak. Sedangkan implementasi antarmuka untuk halaman laporan riwayat hanya menampilkan daftar laporan yang telah dilaporkan oleh pelapor yang sedang login dan status pelaporannya sudah selesai atau sudah ditolak. Informasi yang ditampilkan pada laporan yang sedang berjalan dan laporan riwayat sama dengan informasi yang ditampilkan pada halaman laporan lini masa.

6.3.1.4 Antarmuka Halaman Rincian Laporan

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan halaman rincian laporan pada gambar 5.18 ke dalam wujud tangkapan layar antarmuka. Implementasi antarmuka untuk halaman rincian laporan ditunjukkan pada Gambar 6.6 berikut.



Gambar 6.6 Implementasi Halaman Rincian Laporan

Gambar 6.6 menunjukkan halaman yang menampilkan rincian dari laporan yang tersedia dari halaman laporan lini masa, sedang berjalan atau riwayat. Halaman rincian laporan dapat ditampilkan dengan cara memilih salah satu laporan yang tersedia dari halaman laporan lini masa, sedang berjalan atau riwayat. Informasi rincian laporan yang ditampilkan pada halaman ini antara lain adalah judul laporan, nama pelapor, nama teknisi yang memperbaiki, nama aset, foto laporan, deskripsi laporan, saran tindakan, waktu dilaporkan dan diperbarui, status pelaporan, hasil survey dan hasil akhir tindakan. Untuk lebih menjelaskan perkembangan dari penindak lanjutan laporan, pada awal halaman rincian telah disediakan langkah-langkah status pelaporan yang telah dilakukan oleh pegawai perlengkapan. Status pelaporan yang ditunjukkan pada perkembangan status pelaporan tersebut antara lain:

- a) Diterima
Untuk menjelaskan bahwa laporan keluhan telah diterima oleh pegawai perlengkapan.
- b) Survey
Untuk menjelaskan bahwa laporan keluhan sedang pada tahap disurvey oleh pegawai perlengkapan.
- c) Menunggu
Untuk menjelaskan bahwa laporan keluhan sedang pada tahap menunggu barang pengganti, suku cadang, atau ketersediaan teknisi.
- d) Perbaikan
Untuk menjelaskan bahwa laporan keluhan telah dilakukan disposisi teknisi dan sedang pada tahap diperbaiki oleh teknisi.
- e) Selesai
Untuk menjelaskan bahwa laporan keluhan telah selesai diperbaiki atau ditolak.

6.3.2 Implementasi Antarmuka *Website Operator*

Pada bagian ini merupakan implementasi antarmuka dari *website* untuk *operator*. Implementasi antarmuka yang dijelaskan pada bagian ini terdiri dari penjelasan mengenai antarmuka untuk *login* dan *sign up operator*, halaman daftar laporan dan aset, halaman rincian laporan dan aset serta halaman untuk disposisi teknisi sesuai dengan kebutuhan yang dijelaskan pada perancangan antarmuka pengguna *operator*. Mengenai implementasi antarmuka dari halaman daftar dan rincian dari teknisi dan pelapor akan diwakilkan oleh halaman daftar dan rincian dari laporan dan aset.

6.3.2.1 Implementasi Halaman Login dan Sign Up Operator

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan halaman *login* dan *sign up operator* pada Gambar 5.19 ke dalam wujud tangkapan layar antarmuka. Implementasi antarmuka untuk halaman *login* dari *operator* ditunjukkan pada Gambar 6.7 berikut.

Gambar 6.7 Implementasi Halaman Login Operator

Pada Gambar 6.7 menunjukkan halaman yang digunakan oleh *operator* untuk memasukkan identitas pegawai perlengkapan dan *password* agar dapat *login* ke dalam menu utama *website operator*. Pada bagian bawah tombol *login* terdapat pilihan fungsi untuk menuju halaman *sign up* apabila pegawai perlengkapan belum memiliki akun *operator* yang terdaftar. Implementasi antarmuka untuk halaman *sign up* dari *operator* ditunjukkan pada Gambar 6.8 berikut.

Gambar 6.8 Implementasi Halaman Sign Up Operator

Gambar 6.8 menunjukkan halaman yang berisi *form* yang dapat digunakan pegawai perlengkapan untuk mendaftarkan akun sebagai *operator*. Pada halaman ini pegawai perlengkapan memasukkan NIK atau NIP, nama lengkap, jenis kelamin, jabatan, *username*, *password*, *email* dan nomor telepon. Apabila *form* telah terisi lengkap maka pegawai perlengkapan dapat memilih tombol *sign up* untuk mendaftarkan akun *operator*nya.

6.3.2.2 Implementasi Halaman Daftar Laporan dan Daftar Aset

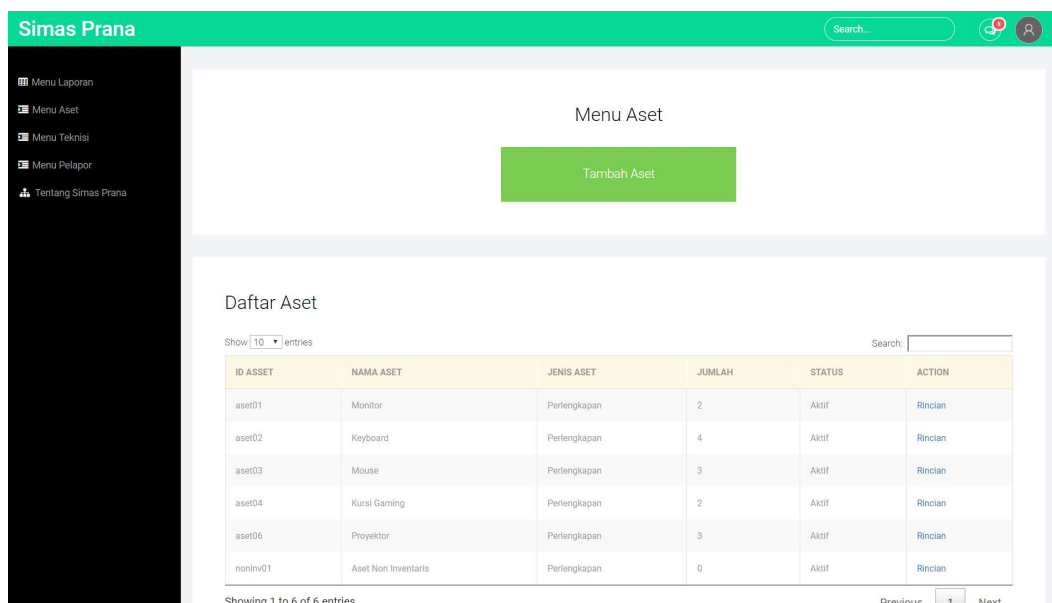
Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan antarmuka *operator* untuk halaman daftar laporan, aset, teknisi dan pelapor pada Gambar 5.20 ke dalam wujud tangkapan layar antarmuka. Pada implementasi halaman daftar ini akan diwakilkan dengan halaman daftar laporan dan daftar aset saja. Implementasi antarmuka untuk halaman daftar laporan dan notifikasi laporan ditunjukkan pada Gambar 6.9 berikut.

The screenshot displays the 'Simas Prana' web application. The top navigation bar is green with the title 'Simas Prana' and a search bar. The left sidebar contains a menu with options: 'Menu Laporan', 'Menu Aset', 'Menu Teknisi', 'Menu Pelapor', and 'Tentang Simas Prana'. The main content area is titled 'Menu Laporan' and features a green 'Tambah Laporan' button. Below this, there is a section titled 'Keluhan Sarana Prasarana' with a table of facility complaints. The table has columns for ID LAPORAN, ID PELAPOR, TEKNIKI, ID ASET, WAKTU DILAPORKAN, JUDUL, and ACTION. A notification panel on the right shows a list of recent reports with details like 'Tembok Pecah', 'Judul', 'Laporan AutoJoss', and 'Judul Kursi'.

ID LAPORAN	ID PELAPOR	TEKNIKI	ID ASET	WAKTU DILAPORKAN	JUDUL	ACTION
1	145150401111063	Fahrudin	aset01	Rabu, 04 April 2018 11:22:46	Monitor Rusak	Rincian
2	198504102012121001	Daud	aset02	Kamis, 05 April 2018 09:37:47	Keyboard Bermasalah	Rincian
3	2016079001051001	Fahrudin	aset01	Kamis, 05 April 2018 09:37:47	Montitornya Mbeleret	Rincian
4	2012068701122101	-	aset02	Sabtu, 07 April 2018 04:58:52	Broken Keyboard	Rincian
5	2012068701122101	-	aset02	Sabtu, 07 April 2018 05:03:22	Broken Keyboard Agan	Rincian
8	2012068701122101	-	aset06	Rabu, 11 April 2018 09:10:05	Proyekitornya Mbeleret	Rincian
9	145150401111063	-	aset01	Rabu, 11 April 2018 11:04:57	Mobile Report	Rincian

Gambar 6.9 Implementasi Halaman Daftar Laporan dan Notifikasi Laporan

Gambar 6.9 menunjukkan halaman yang menyediakan menu untuk tambah laporan kemudian dilanjutkan dengan daftar keluhan sarana prasarana yang disajikan dengan sebuah tabel. Informasi yang ditampilkan pada tabel tersebut terdapat ID laporan, ID pelapor, nama teknisi yang memperbaiki, ID aset, waktu dilaporkan, judul laporan dan fungsi untuk membuka halaman rincian laporan. Selain itu pada bagian *header* halaman tersebut juga disediakan fungsi notifikasi yang menampilkan daftar laporan terbaru beserta dengan informasi judul laporan, foto laporan dan waktu dilaporkannya. Tiap baris laporan pada notifikasi tersebut juga dapat dipilih untuk membuka halaman rincian laporan dari laporan tersebut. Sedangkan implementasi antarmuka untuk halaman daftar aset ditunjukkan pada Gambar 6.10 berikut.



Gambar 6.10 Implementasi Halaman Daftar Aset

Gambar 6.10 menunjukkan halaman yang menyediakan menu untuk tambah aset baru kemudian dilanjutkan dengan daftar aset sarana prasarana yang disajikan dengan sebuah tabel. Informasi yang ditampilkan pada tabel tersebut terdapat ID Aset, nama aset, jenis aset, jumlah aset, status aset dan fungsi untuk membuka halaman rincian dari aset tersebut.

6.3.2.3 Implementasi Halaman Rincian Laporan dan Aset

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan antarmuka *operator* untuk halaman rincian laporan, aset, teknisi dan pelapor pada Gambar 5.21 ke dalam wujud tangkapan layar antarmuka. Pada implementasi halaman rincian ini akan diwakilkan dengan halaman rincian laporan dan rincian aset saja. Implementasi antarmuka untuk halaman rincian laporan ditunjukkan pada Gambar 6.11 berikut.

Gambar 6.11 Implementasi Halaman Rincian Laporan

Gambar 6.11 menunjukkan halaman yang menyediakan *form* rincian dari laporan. *Form* rincian menampilkan informasi-informasi seperti ID laporan, judul laporan, ID dan nama pelapor, ID dan nama teknisi, ID dan nama aset sarana prasarana, waktu dilaporkan, waktu *diupdate*, deskripsi laporan, saran perbaikan, status pelaporan, hasil survey, hasil akhir dan foto laporan. Bagian bawah dari halaman rincian terdapat tombol *submit* untuk menyimpan perubahan data status laporan atau hasil survey dan hasil akhir yang dilakukan oleh *operator* serta tombol *report record* yang berfungsi untuk menampilkan halaman rekam pembaharuan data laporan oleh *operator*. Sedangkan implementasi antarmuka untuk halaman rincian aset ditunjukkan pada Gambar 6.12 berikut.

The screenshot shows the 'Rincian Aset' page in the Simas Prana application. On the left is a dark sidebar with menu items: 'Menu Aset', 'Menu Teknisi', 'Menu Pelapor', and 'Tentang Simas Prana'. The main content area has a green header with the title 'Rincian Aset' and a search bar. Below the header, there are two columns of input fields. The left column contains 'ID Aset' (aset01), 'Nama Aset' (Monitor), 'Lokasi Aset' (F1.1), and 'Foto Aset' (a monitor image). The right column contains 'Jenis Aset' (Perengkapan), 'Jumlah Aset' (2), 'Status Aset' (Aktif), and 'QR Code Aset' (a QR code). At the bottom of the form are two buttons: 'Submit' and 'Download QR'.

Gambar 6.12 Implementasi Halaman Rincian Aset

Gambar 6.12 menunjukkan halaman yang menyediakan *form* rincian dari aset. *Form* rincian menampilkan informasi-informasi seperti ID aset, nama aset, lokasi dari aset, jenis aset, jumlah aset, status aset, foto aset dan kode QR dari aset. Status dari aset terdiri dari aktif dan tidak aktif. Apabila status aset dirubah menjadi tidak aktif, sistem akan menampilkan pesan peringatan bahwa aset yang tidak aktif tidak akan ditampilkan lagi. Kode QR aset berisi kode enkripsi yang mengandung informasi dari ID aset, nama aset, lokasi aset dan jenis aset. Ketika kode QR ini di pindai oleh fitur pindai pada aplikasi *mobile* pelapor maka informasi tersebut akan secara otomatis mengisi informasi tentang aset pada form buat laporan baru.

6.3.2.4 Implementasi Halaman Disposisi Teknisi

Bagian ini merupakan bagian implementasi dari perancangan yang telah dilakukan pada bagian rancangan antarmuka *operator* untuk halaman disposisi teknisi pada Gambar 5.22 ke dalam wujud tangkapan layar antarmuka. Implementasi antarmuka untuk halaman disposisi teknisi ditunjukkan pada Gambar 6.13 berikut.

Simas Prana

Search...

Menu Laporan
Menu Aset
Menu Teknisi
Menu Pelapor
Tentang Simas Prana

Report Disposition

ID Report: 30

ID Asset: noninv01

Title: Laporan Platon

Description: Platonya jebol

Suggestion: Diperbaiki

Technician List:

Show 10 entries

ID TECHNICIAN	FULLNAME	STATUS	DISPOSITION
daud	Daud Siewo Sarjono	Tersedia	Give Order
fahrudin	Fahrudin Rachmanawan Saputra, S.E.	Tersedia	Give Order
ferix	Ferix Panji Andrianto, S.ST	Tersedia	Give Order
pranoe	Pranoe Taterpiga, S.H	Tersedia	Give Order

Showing 1 to 4 of 4 entries

Previous 1 Next

Copyright © 2018 Modern. All Rights Reserved | Design by Danniart Ganteng

Gambar 6.13 Implementasi Halaman Disposisi Teknisi

Gambar 6.13 menunjukkan halaman yang menyediakan *form* ringkas dari laporan dan tabel daftar teknisi yang telah terdaftar. *Form* ringkas dari laporan menampilkan informasi-informasi seperti ID laporan, ID aset, judul laporan, deskripsi laporan dan saran perbaikan. Sedangkan tabel daftar teknisi menampilkan informasi-informasi seperti ID teknisi, nama lengkap, status teknisi dan fungsi untuk memberi disposisi. Status dari teknisi terdiri dari tersedia, sedang bekerja dan sedang keluar. Apabila status teknisi sedang bekerja atau sedang keluar maka teknisi tersebut tidak dapat diberikan disposisi untuk laporan keluhan tersebut.

BAB 7 PENGUJIAN DAN ANALISIS HASIL

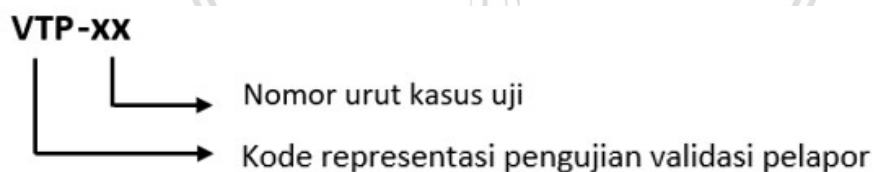
Pengujian merupakan representasi dari Tahap Pengujian pada *Waterfall Model* yang mana merupakan proses untuk memeriksa apakah solusi yang ditawarkan perangkat lunak tersebut telah memenuhi persyaratan dan spesifikasi yang telah disepakati di awal serta memenuhi tujuannya. Pembahasan pada bab pengujian dan analisis hasil ini diuraikan menjadi pengujian *black-box*, pengujian *usability* dan pengujian perbandingan waktu.

7.1 Pengujian *Black-Box*

Pengujian *black-box* merupakan pengujian yang dilakukan pada antarmuka dari perangkat lunak dan berfokus kepada kebutuhan fungsional dari perangkat lunak tersebut. Pengujian ini berusaha untuk mencari kesalahan mengenai fungsi yang salah atau hilang, kesalahan antarmuka, kesalahan dalam struktur data serta kesalahan perilaku atau kinerja sistem. Pengujian pada bagian ini dilakukan dengan mendefinisikan kasus uji yang merujuk pada *use case scenario* yang telah didefinisikan pada Tahap Analisis Kebutuhan. Kemudian setiap hasil dari pengujian yang dilakukan akan dibandingkan dengan hasil yang telah didefinisikan pada *use case scenario*. Pengujian *black-box* pada bagian ini dibagi menjadi pengujian *black-box* aplikasi pelapor dan pengujian *black-box website operator* menggunakan pengujian validasi.

7.1.1 Pengujian Validasi Aplikasi Pelapor

Pengujian validasi pada bagian ini dilakukan untuk aplikasi yang digunakan oleh pelapor khususnya dilakukan pada fitur untuk mengirim laporan dan melihat laporan. Berikut pada Gambar 7.1 merupakan aturan penomoran yang diterapkan untuk pengujian validasi aplikasi pelapor.



Gambar 7.1 Aturan penomoran pengujian validasi pelapor

7.1.1.1 Pengujian Validasi Mengirim Laporan

Pada Tabel 7.1 menjelaskan mengenai kasus uji dari pengujian validasi mengirim laporan dengan kode VTP-01. Kasus uji VTP-01 merupakan pengujian validasi yang merujuk pada *use case* dengan kode UCP-03 Mengirim Laporan. Berdasarkan pada Tabel 7.1 hasil status pengujian validasi untuk mengirim laporan menunjukkan hasil *valid*.

Tabel 7.1 Kasus Uji Pengujian Validasi Mengirim Laporan

Kode Pengujian	VTP-01
Kode Use Case	UCP-03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat mengirimkan laporan keluhan yang telah dimasukkan oleh pelapor.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah teridentifikasi sebagai pelapor 2. Penguji memilih fungsi tambah laporan baru pada menu navigasi 3. Penguji mengisi <i>form</i> yang diperlukan untuk membuat laporan baru 4. Penguji memilih fitur pindai kode QR untuk mengisi informasi sarana prasarana yang dikeluhkan 5. Penguji memilih fitur kamera untuk menambahkan foto laporan keluhan 6. Penguji memilih fungsi <i>submit</i> untuk mengirimkan keluhan yang telah dilaporkan.
Hasil yang Diharapkan	Sistem berhasil mengirimkan laporan keluhan kepada <i>operator</i> dan kembali diarahkan ke halaman menu utama aplikasi.
Hasil Pengujian	Sistem berhasil mengirimkan laporan keluhan kepada <i>operator</i> dan kembali diarahkan ke halaman menu utama aplikasi.
Status Uji	<i>Valid</i>

7.1.1.2 Pengujian Validasi Melihat Laporan

Pada Tabel 7.2, Tabel 7.3 dan Tabel 7.4 secara berurutan menjelaskan mengenai kasus uji dari pengujian validasi melihat laporan dengan kode VTP-02, VTP-03 dan VTP-04. Kasus uji dengan kode VTP-02 merujuk pada *use case* dengan kode UCP-04 Melihat Laporan Lini Masa. Kasus uji dengan kode kode VTP-03 merujuk pada *use case* dengan kode UCP-05 Melihat Laporan yang Sedang Ditangani. Sedangkan kasus uji dengan kode kode VTP-04 merujuk pada *use case* dengan kode UCP-06 Melihat Laporan Riwayat. Berdasarkan pada Tabel 7.2, Tabel 7.3 dan Tabel 7.4 hasil status pengujian validasi untuk melihat laporan menunjukkan hasil *valid*.

Tabel 7.2 Kasus Uji Pengujian Validasi Melihat Laporan Lini Masa

Kode Pengujian	VTP-02
Kode Use Case	UCP-04
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan halaman kumpulan keluhan-keluhan yang telah dilaporkan oleh pelapor lain
Prosedur Uji	1. Penguji telah teridentifikasi sebagai pelapor 2. Penguji memilih fungsi tampilkan daftar laporan lini masa pada menu navigasi
Hasil yang Diharapkan	Sistem berhasil menampilkan laporan keluhan-keluhan yang telah dilaporkan oleh pelapor lain.
Hasil Pengujian	Sistem berhasil menampilkan laporan keluhan-keluhan yang telah dilaporkan oleh pelapor lain.
Status Uji	<i>Valid</i>

Tabel 7.3 Kasus Uji Pengujian Validasi Melihat Laporan yang Sedang Ditangani

Kode Pengujian	VTP-03
Kode Use Case	UCP-05
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan halaman kumpulan keluhan yang telah dilaporkan oleh pelapor yang sedang <i>login</i> yang belum diselesaikan.
Prosedur Uji	1. Penguji telah teridentifikasi sebagai pelapor 2. Penguji memilih fungsi tampilkan daftar laporan yang sedang ditangani pada menu navigasi
Hasil yang Diharapkan	Sistem berhasil menampilkan laporan keluhan yang telah dilaporkan oleh pelapor yang sedang <i>login</i> yang belum diselesaikan.
Hasil Pengujian	Sistem berhasil menampilkan laporan keluhan yang telah dilaporkan oleh pelapor yang sedang <i>login</i> yang belum diselesaikan.
Status Uji	<i>Valid</i>

Tabel 7.4 Kasus Uji Pengujian Validasi Melihat Laporan Riwayat

Kode Pengujian	VTP-04
Kode Use Case	UCP-06
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan halaman kumpulan keluhan yang telah dilaporkan oleh pelapor yang sedang <i>login</i> yang sudah diselesaikan.
Prosedur Uji	1. Penguji telah teridentifikasi sebagai pelapor 2. Penguji memilih fungsi tampilkan daftar laporan riwayat pada menu navigasi
Hasil yang Diharapkan	Sistem berhasil menampilkan laporan keluhan yang telah dilaporkan oleh pelapor yang sedang <i>login</i> yang sudah diselesaikan.
Hasil Pengujian	Sistem berhasil menampilkan laporan keluhan yang telah dilaporkan oleh pelapor yang sedang <i>login</i> yang sudah diselesaikan.
Status Uji	<i>Valid</i>

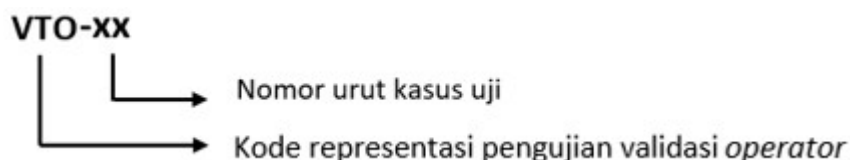
Berdasarkan dari hasil pengujian validasi aplikasi pelapor yang telah dilakukan, berikut pada Tabel 7.5 merupakan rangkuman hasil dari pengujian validasi untuk aplikasi *mobile* pelapor yang menjelaskan bahwa nomor kode uji VTP-01 hingga VTP-04 adalah valid. Dari hasil yang didapatkan tersebut maka dapat disimpulkan bahwa perangkat lunak yang dikembangkan dapat berjalan sesuai dengan hasil yang diharapkan.

Tabel 7.5 Tabel Hasil Pengujian Validasi Aplikasi Mobile Pelapor

Nomor Kode Uji	Hasil
VTP-01	Valid
VTP-02	Valid
VTP-03	Valid
VTP-04	Valid

7.1.2 Pengujian Validasi *Website Operator*

Pengujian validasi pada bagian ini dilakukan untuk *website* yang digunakan oleh *operator* khususnya dilakukan pada fitur untuk menerima laporan dan menindaklanjuti laporan. Berikut pada Gambar 7.2 merupakan aturan penomoran yang diterapkan untuk pengujian validasi *website operator*.



Gambar 7.2 Aturan penomoran pengujian validasi operator

7.1.2.1 Pengujian Validasi Menerima Laporan

Pada Tabel 7.6 menjelaskan mengenai kasus uji dari pengujian validasi menerima laporan dengan kode VTO-01. Kasus uji VTO-01 merujuk pada *use case* dengan kode UCO-03 Melihat Laporan. Berdasarkan pada Tabel 7.6 hasil status pengujian validasi untuk menerima laporan menunjukkan hasil *valid*.

Tabel 7.6 Kasus Uji Pengujian Validasi Menerima Laporan

Kode Pengujian	VTO-01
Kode <i>Use Case</i>	UCO-03
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menampilkan halaman bagi <i>operator</i> untuk melihat laporan masuk yang dilaporkan oleh pelapor.
Prosedur Uji	<p>Skenario utama:</p> <ol style="list-style-type: none"> 1. Penguji telah teridentifikasi sebagai <i>operator</i> 2. Penguji memilih fungsi untuk membuka halaman seluruh laporan 3. Penguji memilih fungsi untuk lihat rincian laporan tertentu <p>Skenario alternatif:</p> <ol style="list-style-type: none"> 1. Penguji telah teridentifikasi sebagai <i>operator</i> 2. Penguji memilih fungsi untuk lihat notifikasi masuk 3. Penguji memilih fungsi lihat rincian laporan tertentu
Hasil yang Diharapkan	Sistem berhasil menampilkan halaman untuk melihat laporan masuk yang dilaporkan oleh pelapor.
Hasil Pengujian	Sistem berhasil menampilkan halaman untuk melihat laporan masuk yang dilaporkan oleh pelapor.
Status Uji	<i>Valid</i>

7.1.2.2 Pengujian Validasi Menindaklanjuti Laporan

Pada Tabel 7.7 menjelaskan mengenai kasus uji dari pengujian validasi menindaklanjuti laporan dengan kode VTO-02. Kasus uji VTO-02 merujuk pada *use case* dengan kode UCO-04 Menindaklanjuti Laporan. Berdasarkan pada Tabel 7.7 hasil status pengujian validasi untuk menindaklanjuti laporan menunjukkan hasil *valid*.

Tabel 7.7 Kasus Uji Pengujian Validasi Menindaklanjuti Laporan

Kode Pengujian	VTO-02
Kode Use Case	UCO-04
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat mengubah status laporan dan memasukkan keterangan dari perubahan status tindak lanjut.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji telah teridentifikasi sebagai <i>operator</i> 2. Penguji merubah status tindak lanjut menjadi sedang disurvey dan memasukkan hasil survey 3. Penguji merubah status tindak lanjut menjadi sedang diperbaiki 4. Penguji melakukan disposisi teknisi 5. Penguji merubah status tindak lanjut menjadi selesai 6. Penguji memasukkan hasil akhir perbaikan
Hasil yang Diharapkan	Sistem berhasil melakukan perubahan status untuk menindaklanjuti laporan keluhan yang masuk.
Hasil Pengujian	Sistem berhasil melakukan perubahan status untuk menindaklanjuti laporan keluhan yang masuk.
Status Uji	<i>Valid</i>

Berdasarkan dari hasil pengujian validasi website *operator* yang telah dilakukan, berikut pada Tabel 7.8 merupakan rangkuman hasil dari pengujian validasi untuk website *operator* yang menjelaskan bahwa nomor kode uji VTO-01 dan VTO-02 adalah valid. Dari hasil yang didapatkan tersebut maka dapat disimpulkan bahwa perangkat lunak yang dikembangkan dapat berjalan sesuai dengan hasil yang diharapkan.

Tabel 7.8 Tabel Hasil Pengujian Validasi Website Operator

Nomor Kode Uji	Hasil
VTO-01	Valid
VTO-02	Valid

7.2 Pengujian *Usability*

Pengujian *Usability* atau *Usability Testing* merupakan pengujian yang melibatkan perwakilan pengguna dari perangkat lunak yang akan diuji dengan memberikan *task-task* yang perlu dilakukan oleh perwakilan pengguna. Setelah *task-task* diberikan kepada perwakilan pengguna, akan dilakukan pengukuran perihal *effectiveness*, *efficiency* dan *satisfaction* yang mengacu kepada standar metrik *usability* ISO/IEC 9126-4. Pada pengujian ini akan dibagi menjadi 3 bagian yaitu masing-masing untuk mengukur metrik *effectiveness*, *efficiency* dan *satisfaction*.

7.2.1 Metrik *Effectiveness*

Pada bagian ini akan dilakukan pengukuran metrik *effectiveness* dari *usability testing* yang diawali dengan memberikan *task-task* yang perlu dilakukan kepada perwakilan pengguna. Kemudian pengukuran metrik *effectiveness* dilakukan dengan cara menghitung nilai *effectiveness* dari *task-task* yang berhasil dilakukan oleh tiap perwakilan pengguna dan selanjutnya nilai *effectiveness* yang telah didapatkan akan dilakukan penghitungan rata-rata dengan seluruh perwakilan pengguna untuk mendapatkan nilai *effectiveness* rata-rata seluruh perwakilan pengguna. Pada pengukuran metrik *effectiveness* dari *usability testing* ini akan dibagi menjadi pengukuran metrik *effectiveness* pengguna pelapor dan pengukuran metrik *effectiveness* pengguna operator.

7.2.1.1 Pengukuran Metrik *Effectiveness* Pengguna Pelapor

Pada Tabel 7.9 merupakan tabel daftar *task* yang akan diujikan kepada perwakilan pengguna pelapor yang akan ditandai dengan kode *task*, nama *task*, kode *use case* yang menjadi acuan, dan langkah skenario yang akan dilakukan. Kode dari setiap *task* yang diberikan kepada pengguna pelapor akan diwakilkan dengan kode TP-00. TP merupakan singkatan dari *Task Pelapor*, sedangkan angka 00 merupakan penomoran dari kode *task* tersebut.

Tabel 7.9 Daftar Task Usability Testing Pengguna Pelapor

Kode Task	Nama Task	Kode Use Case	Skenario Pengujian
TP-01	Mendaftarkan Akun	UCP-01	<ol style="list-style-type: none"> 1. Pengguna memilih pilihan menu pada halaman login untuk membuka halaman daftar akun. 2. Pengguna mengisi semua <i>form</i> masukan yang diperlukan serta mengunggah foto kartu identitas untuk mendaftarkan akun sebagai Mahasiswa. 3. Pengguna memilih tombol <i>submit</i> untuk mendaftarkan akun.

Kode Task	Nama Task	Kode Use Case	Skenario Pengujian
TP-02	Melakukan <i>Login</i>	UCP-02	<ol style="list-style-type: none"> Pengguna memasukkan data NIM dan <i>password</i> dari akun yang telah berhasil terdaftar ke dalam <i>form login</i> yang tersedia. Pengguna memilih tombol <i>login</i> untuk mendapatkan akses <i>login</i>.
TP-03	Mengirim Laporan	UCP-03	<ol style="list-style-type: none"> Pengguna melakukan <i>swipe</i> ke kanan pada bagian layar sebelah kiri untuk memunculkan daftar menu, kemudian memilih menu laporan baru. Pengguna mengisi <i>form</i> yang telah disediakan untuk membuat laporan keluhan baru untuk sarana prasarana inventaris. Pengguna melakukan pindai kode QR untuk mendapatkan informasi mengenai sarana prasarana yang dilaporkan. Pengguna mengambil dan mengunggah foto dari sarana prasarana yang dilaporkan. Pengguna memilih tombol <i>submit</i> untuk mengirimkan laporan.
TP-04	Melihat Laporan yang Sedang Ditangani	UCP-05	<ol style="list-style-type: none"> Pengguna melakukan <i>swipe</i> ke kanan pada bagian layar sebelah kiri untuk memunculkan daftar menu, kemudian memilih menu Laporan yang Sedang Ditangani. Pengguna memilih salah satu laporan keluhan untuk melihat informasi rincian lengkap dari laporan keluhan tersebut.
TP-05	Melihat Laporan Riwayat	UCP-06	<ol style="list-style-type: none"> Pengguna melakukan <i>swipe</i> ke kanan pada bagian layar sebelah kiri untuk memunculkan daftar menu, kemudian memilih menu Laporan Riwayat. Pengguna memilih salah satu laporan keluhan untuk melihat informasi

Kode Task	Nama Task	Kode Use Case	Skenario Pengujian
			rincian lengkap dari laporan keluhan tersebut.
TP-06	Melihat Laporan Lini Masa	UCP-04	<ol style="list-style-type: none"> Pengguna melakukan <i>swipe</i> ke kanan pada bagian layar sebelah kiri untuk memunculkan daftar menu, kemudian memilih menu Laporan Lini Masa. Pengguna memilih salah satu laporan keluhan untuk melihat informasi rincian lengkap dari laporan keluhan tersebut.
TP-07	Melakukan Logout	UCP-07	<ol style="list-style-type: none"> Pengguna melakukan <i>swipe</i> ke kanan pada bagian layar sebelah kiri untuk memunculkan daftar menu, kemudian memilih menu Logout.

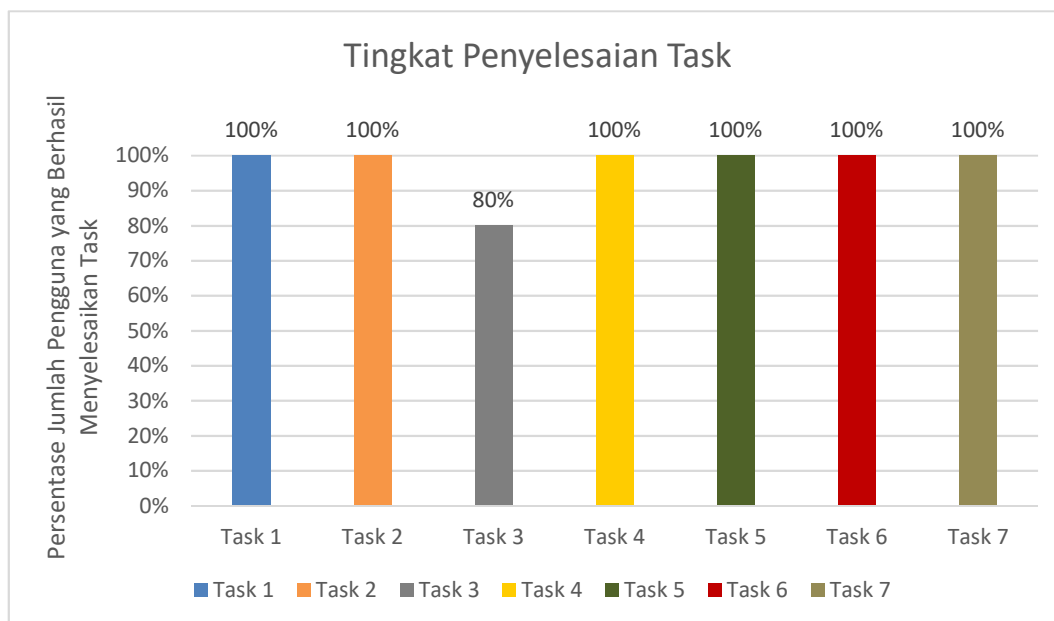
Pada Tabel 7.10 merupakan tabel hasil keberhasilan pengujian *usability testing* yang dilakukan oleh perwakilan pengguna pelapor untuk mengukur metrik *effectiveness*. Pada Tabel 7.10, nilai yang diisikan ke dalam kolom-kolom dari daftar kode *task* merupakan nilai biner '1' jika responden berhasil menyelesaikan suatu *task* dan '0' jika tidak. Sedangkan nilai yang akan diisikan ke dalam kolom *Effectiveness* adalah nilai persentase hasil dari perhitungan *effectiveness* dengan menggunakan rumus yang telah dijelaskan pada Rumus 2.1.

Tabel 7.10 Hasil Keberhasilan Task Pengguna Pelapor

Pengguna	Kode Task							Effectiveness
	TP-01	TP-02	TP-03	TP-04	TP-05	TP-06	TP-07	
A	1	1	1	1	1	1	1	100%
B	1	1	1	1	1	1	1	100%
C	1	1	1	1	1	1	1	100%
D	1	1	1	1	1	1	1	100%
E	1	1	0	1	1	1	1	85,7%
Rata-Rata								97,14%

Untuk mempermudah dalam pembacaan hasil pengukuran metrik *effectiveness* dari Tabel 7.10, berikut pada Gambar 7.3 disajikan berupa grafik batang tingkat penyelesaian *task* pengguna pelapor. Pada sumbu x menjelaskan daftar *task* yang diberikan kepada perwakilan pengguna pelapor dan pada sumbu y menjelaskan persentase jumlah pengguna yang berhasil menyelesaikan masing-

masing *task* dari sumbu x. Berdasarkan hasil dari Tabel 7.10 dan Gambar 7.3 berikut didapatkan satu pengguna yang tidak berhasil dalam melakukan satu *task* yang diberikan, yaitu pada *Task 3*. Hal ini diakibatkan karena ketika melakukan pengujian untuk mengirimkan laporan oleh pengguna tersebut, terjadi kesalahan pemilihan fungsi pada aplikasi pelapor yang menyebabkan kegagalan dalam melakukan pemindaian kode QR.



Gambar 7.3 Grafik Batang Hasil Tingkat Penyelesaian Pengguna Pelapor

7.2.1.2 Pengukuran Metrik *Effectiveness* Pengguna *Operator*

Berikut pada Tabel 7.11 merupakan tabel daftar *task* yang akan diujikan kepada perwakilan pengguna *operator* yang akan ditandai dengan kode *task*, nama *task*, kode *use case* yang menjadi acuan, dan langkah skenario yang akan dilakukan. Kode dari setiap *task* yang diberikan kepada pengguna *operator* akan diwakilkan dengan kode TO-00. TO merupakan singkatan dari *Task operator*, sedangkan angka 00 merupakan penomoran dari kode *task* tersebut.

Tabel 7.11 Daftar Task Usability Testing Pengguna *Operator*

Kode Task	Nama Task	Kode Use Case	Skenario Pengujian
TO-01	Mendaftarkan Akun	UCO-01	<ol style="list-style-type: none"> Pengguna memilih pilihan menu pada halaman login untuk membuka halaman daftar akun. Pengguna mengisi semua <i>form</i> masukan yang diperlukan untuk mendaftarkan akun sebagai <i>Operator</i>. Pengguna memilih tombol <i>submit</i> untuk mendaftarkan akun <i>operator</i>.

Kode Task	Nama Task	Kode Use Case	Skenario Pengujian
TO-02	Melakukan <i>Login</i>	UCO-02	<ol style="list-style-type: none"> 1. Pengguna memasukkan data NIK dan <i>password</i> dari akun <i>operator</i> yang telah berhasil terdaftar ke dalam <i>form login</i> yang tersedia. 2. Pengguna memilih tombol <i>login</i> untuk mendapatkan akses <i>login</i>.
TO-03	Melihat Notifikasi Laporan	UCO-03	<ol style="list-style-type: none"> 1. Pengguna memilih <i>icon</i> notifikasi pada sebelah kanan atas untuk membuka daftar laporan masuk. 2. Pengguna memilih salah satu laporan yang ditampilkan pada notifikasi untuk membuka halaman rincian laporan yang dipilih.
TO-04	Melihat Seluruh Laporan	UCO-03	<ol style="list-style-type: none"> 1. Pengguna memilih <i>menu</i> laporan di sebelah kiri layar untuk menampilkan seluruh daftar laporan masuk. 2. Pengguna memilih salah satu laporan yang ditampilkan pada daftar laporan masuk untuk membuka halaman rincian laporan yang dipilih.
TO-05	Menindak Lanjuti Laporan	UCO-04	<ol style="list-style-type: none"> 1. Pengguna membuka halaman rincian laporan yang akan ditindaklanjuti. 2. Pengguna merubah status tindak lanjut menjadi sedang disurvei. 3. Pengguna memasukkan hasil survey laporan ke dalam <i>form</i> yang telah disediakan. 4. Pengguna memilih tombol <i>submit</i> untuk menyimpan perubahan data. 5. Pengguna merubah status tindak lanjut menjadi sedang diperbaiki. 6. Pengguna melakukan disposisi teknisi dengan cara memilih teknisi yang akan diberikan disposisi tugas. 7. Pengguna memilih tombol <i>submit</i> untuk menyimpan pemberian disposisi. 8. Pengguna merubah status tindak lanjut menjadi selesai.

Kode Task	Nama Task	Kode Use Case	Skenario Pengujian
			9. Pengguna memasukkan hasil akhir perbaikan ke dalam <i>form</i> yang telah disediakan. 10. Pengguna memilih tombol <i>submit</i> untuk menyimpan perubahan data.
TO-06	Melakukan <i>Logout</i>	UCO-13	1. Pengguna memilih ikon <i>setting</i> akun pada bagian kanan atas layar, kemudian memilih fungsi <i>logout</i> .

Berikut Tabel 7.12 merupakan tabel hasil keberhasilan pengujian *usability testing* yang dilakukan oleh perwakilan pengguna *operator* untuk mengukur metrik *effectiveness*. Pada Tabel 7.12, nilai yang akan diisikan ke dalam kolom-kolom dari daftar kode *task* merupakan nilai biner '1' jika responden berhasil menyelesaikan suatu *task* dan '0' jika tidak. Sedangkan nilai yang akan diisikan ke dalam kolom *Effectiveness* adalah nilai persentase hasil dari perhitungan *effectiveness* dengan menggunakan rumus yang telah dijelaskan pada Rumus 2.1.

Tabel 7.12 Hasil Keberhasilan Task Pengguna Operator

Pengguna	Kode Task						Effectiveness
	TP-01	TP-02	TP-03	TP-04	TP-05	TP-06	
Pengguna Operator	1	1	1	1	1	1	100%
Rata-Rata							100%

7.2.2 Metrik *Efficiency*

Pada bagian ini akan dilakukan pengukuran metrik *efficiency* dari *usability testing* yang diawali dengan mencatat dan mengukur waktu yang diperlukan oleh pengguna untuk menyelesaikan masing-masing *task-task* yang perlu dilakukan oleh pengguna saat pengukuran metrik *effectiveness*. Kemudian pengukuran metrik *efficiency* dilakukan dengan cara menghitung *time based efficiency* (TBE) dari *task-task* yang perlu dilakukan pengguna untuk masing-masing pengguna dan selanjutnya akan dihitung TBE dari *task-task* untuk keseluruhan pengguna. Pada pengukuran metrik *efficiency* dari *usability testing* ini akan dibagi menjadi pengukuran metrik *efficiency* pengguna pelapor dan pengukuran metrik *efficiency* pengguna operator.

7.2.2.1 Pengukuran Metrik *Efficiency* Pengguna Pelapor

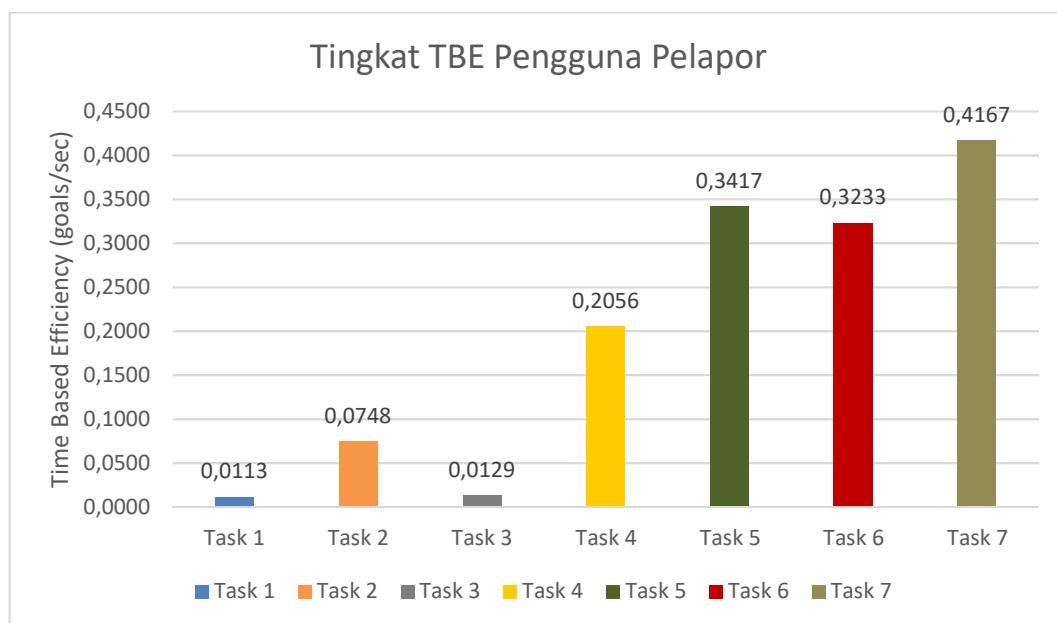
Berikut pada Tabel 7.13 merupakan tabel hasil pengukuran waktu yang diperlukan oleh pengguna pelapor untuk menyelesaikan masing-masing *task-task* yang perlu dilakukan, baik *task* tersebut berhasil diselesaikan maupun tidak. Pada Tabel 7.13 berikut, nilai yang akan diisikan ke dalam kolom-kolom dari daftar kode

task merupakan waktu pengerjaan *task* dalam satuan detik. Sedangkan nilai yang akan diisikan ke dalam kolom TBE adalah nilai *task* yang selesai dilakukan per detik dengan menggunakan rumus yang telah dijelaskan pada Rumus 2.2.

Tabel 7.13 Hasil Pengukuran Waktu Task Pengguna Pelapor

Pengguna	Kode Task							TBE tiap Pengguna	TBE seluruh Pengguna
	TP-01	TP-02	TP-03	TP-04	TP-05	TP-06	TP-07		
A	76	15	53	4	2	2	3	0.240 goals/sec	0.198 goals / sec
B	95	12	55	6	4	3	2	0.195 goals/sec	
C	97	14	68	6	3	3	2	0.204 goals/sec	
D	109	19	79	9	8	5	4	0.109 goals/sec	
E	75	10	95	3	2	4	2	0.242 goals/sec	
TBE tiap Task	0.011 goals/sec	0.075 goals/sec	0.013 goals/sec	0.206 goals/sec	0.342 goals/sec	0.323 goals/sec	0.417 goals/sec		
	0.198 goals/sec								

Untuk mempermudah dalam pembacaan hasil pengukuran metrik *efficiency* dari Tabel 7.13, berikut pada Gambar 7.4 disajikan berupa grafik batang tingkat TBE dari pengguna pelapor. Pada sumbu x menjelaskan daftar *task* yang diberikan kepada perwakilan pengguna pelapor dan pada sumbu y menjelaskan persentase TBE masing-masing *task* dari sumbu x.



Gambar 7.4 Grafik Batang Tingkat TBE Pengguna Pelapor

7.2.2.2 Pengukuran Metrik *Efficiency* Pengguna *Operator*

Berikut pada Tabel 7.14 merupakan tabel hasil pengukuran waktu yang diperlukan oleh pengguna *operator* untuk menyelesaikan masing-masing *task-task* yang perlu dilakukan, baik task tersebut berhasil diselesaikan maupun tidak. Pada Tabel 7.14 berikut, nilai yang akan diisikan ke dalam kolom-kolom dari daftar kode *task* merupakan waktu pengerjaan *task* dalam satuan detik. Sedangkan nilai yang akan diisikan ke dalam kolom TBE adalah nilai task yang selesai dilakukan per detik dengan menggunakan rumus yang telah dijelaskan pada Rumus 2.2.

Tabel 7.14 Hasil Pengukuran Waktu Task Pengguna *Operator*

Pengguna	Kode Task						TBE
	TP-01	TP-02	TP-03	TP-04	TP-05	TP-06	
Pengguna Operator	30	5	5	8	129	3	0,150 <i>goals/sec</i>

7.2.3 Metrik *Satisfaction*

Pada bagian ini akan dilakukan pengukuran metrik *satisfaction* dari *usability testing* yang diawali dengan memberikan kuesioner *task level satisfaction* yang berupa pertanyaan *Single Ease Question* (SEQ) kepada pengguna setelah setiap *task* selesai dilakukan. Pertanyaan yang terdapat pada SEQ terdiri dari pertanyaan mengenai seberapa sulit atau mudah untuk menyelesaikan *task* yang telah diberikan dan pengguna dapat menjawabnya dengan menggunakan poin-poin skala *likert* yang telah disediakan. Kemudian pengukuran metrik *satisfaction* dilakukan dengan cara menghitung hasil dari kuesioner menggunakan skala *likert* tersebut. Pada pengukuran metrik *satisfaction* dari *usability testing* ini akan dibagi menjadi pengukuran metrik *satisfaction* pengguna pelapor dan pengukuran metrik *satisfaction* pengguna operator.

7.2.3.1 Metrik *Satisfaction* Pengguna Pelapor

Berikut pada Tabel 7.15 merupakan tabel hasil kuesioner *task level satisfaction* yang diberikan kepada perwakilan pengguna pelapor yang ditandai dengan daftar pertanyaan, kode *task* acuan dan poin-poin skala *likert*. Satu pertanyaan yang diberikan akan mengacu ke satu *task* yang diwakilkan dengan kode *task* acuan. Sedangkan untuk hasil kuesioner *task level satisfaction* dari masing-masing responden pelapor akan dilampirkan pada bagian Lampiran C.

Tabel 7.15 Kuesioner Task Level Satisfaction Pengguna Pelapor

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai pelapor mudah untuk dilakukan?	TP-01	0	0	1	3	1
2	Apakah proses untuk melakukan <i>login</i> sebagai pelapor mudah untuk dilakukan?	TP-02	0	0	0	0	5
3	Apakah proses untuk membuat dan mengirimkan laporan keluhan sarana prasarana	TP-03	0	0	2	3	0

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	mudah untuk dilakukan?						
4	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan yang sedang ditangani mudah untuk dilakukan?	TP-04	0	0	0	2	3
5	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan riwayat mudah untuk dilakukan?	TP-05	0	0	1	0	4
6	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan lini masa mudah untuk dilakukan?	TP-06	0	0	0	0	5
7	Apakah proses untuk melakukan <i>logout</i> mudah untuk dilakukan?	TP-07	0	0	0	2	3
Total			0	0	4	10	21

Perhitungan skala *likert* dari hasil kuesioner *task level satisfaction* pengguna pelapor diawali dengan menghitung total nilai berdasarkan jawaban dari 5 responden pelapor yang telah terpilih.

$$A = 21 \times 5 = 105$$

$$B = 10 \times 4 = 40$$

$$C = 4 \times 3 = 12$$

$$D = 0 \times 2 = 0$$

$$E = 0 \times 1 = 0$$

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 105 + 40 + 12 + 0 + 0 = 157$$

Kemudian perhitungan skala *likert* dari kuesioner pengujian *task level satisfaction* pengguna pelapor dilanjutkan dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 5 \times 7 = 175$$

Selanjutnya perhitungan skala *likert* dari kuesioner *task level satisfaction* pengguna pelapor diakhiri dengan menghitung nilai persentase menggunakan rumus index.

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (157 \div 175) \times 100\% = 89,7\% \text{ (Sangat Setuju)}$$

7.2.3.2 Metrik Satisfaction Pengguna Operator

Berikut pada Tabel 7.16 merupakan tabel hasil kuesioner *task level satisfaction* yang diberikan kepada perwakilan pengguna *operator* yang ditandai dengan daftar pertanyaan, kode *task* acuan dan poin-poin skala *likert*. Satu pertanyaan yang diberikan akan mengacu ke satu *task* yang diwakilkan dengan kode *task* acuan.

Tabel 7.16 Kuesioner Task Level Satisfaction Pengguna Operator

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai <i>operator</i> mudah untuk dilakukan?	TO-01	0	0	0	1	0
2	Apakah proses untuk melakukan <i>login</i> sebagai <i>operator</i> mudah untuk dilakukan?	TO-02	0	0	0	1	0
3	Apakah proses untuk mendapatkan	TO-03	0	0	1	0	0

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	dan melihat notifikasi laporan keluhan sarana prasarana mudah untuk dilakukan?						
4	Apakah proses untuk mendapatkan dan melihat seluruh laporan keluhan sarana prasarana mudah untuk dilakukan?	TO-04	0	0	0	1	0
5	Apakah proses untuk merubah status laporan keluhan sarana prasarana dan melakukan disposisi teknisi mudah untuk dilakukan?	TO-05	0	0	0	1	0
6	Apakah proses untuk melakukan <i>logout</i> dari sistem mudah untuk dilakukan?	TO-06	0	0	0	1	0
Total			0	0	1	5	0

Perhitungan skala *likert* dari hasil kuesioner *task level satisfaction* pengguna *operator* diawali dengan menghitung total nilai berdasarkan jawaban dari 1 responden pegawai perlengkapan yang bertanggung jawab terhadap bagian pelaporan dan pemeliharaan sarana prasarana.

$$A = 0 \times 5 = 0$$

$$B = 5 \times 4 = 20$$

$$C = 1 \times 3 = 3$$

$$D = 0 \times 2 = 0$$

$$E = 0 \times 1 = 0$$

$$\text{Total Nilai} = A + B + C + D + E$$

$$\text{Total Nilai} = 0 + 20 + 3 + 0 + 0 = 23$$

Kemudian perhitungan skala *likert* dari hasil kuesioner *task level satisfaction* pengguna *operator* dilanjutkan dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 1 \times 6 = 30$$

Selanjutnya perhitungan skala *likert* dari hasil kuesioner *task level satisfaction* pengguna *operator* diakhiri dengan menghitung nilai persentase menggunakan rumus index.

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (23 \div 30) \times 100\% = 76\% \text{ (Setuju)}$$

7.3 Pengujian Perbandingan Waktu

Pengujian Perbandingan Waktu dilakukan untuk melihat hasil perbedaan waktu yang diperlukan oleh pelapor untuk melaporkan keluhan sarana prasarana dan pegawai perlengkapan untuk menindaklanjuti keluhan sarana prasarana yang telah dilaporkan pada proses bisnis sistem yang saat ini sedang berjalan (*as-is*) dengan proses bisnis sistem yang akan diajukan (*to-be*). Proses bisnis *as-is* yang akan dibandingkan pada bagian ini adalah menggunakan proses bisnis pelaporan keluhan sarana prasarana melalui E-Complaint UB seperti yang telah digambarkan di Bab 4 pada Gambar 4.2 dan telah dijelaskan pada Tabel 4.3 dan Tabel 4.4 mengenai hasil analisis waktu penyelesaian aktifitasnya. Pengujian Perbandingan Waktu pada bagian ini akan dipaparkan dalam sebuah tabel yang berisi aktifitas-aktifitas yang dilakukan oleh pengguna pelapor dan pengguna *operator*, mengacu kepada aktifitas yang terdapat pada proses bisnis *to-be* yang telah dibuat sebelumnya beserta waktu pengerjaan aktifitas tersebut oleh para pengguna.

7.3.1 Pengujian Waktu Aktifitas To-Be

Berikut pada Tabel 7.17 menunjukkan hasil pengujian waktu untuk aktifitas-aktifitas pelaporan keluhan sarana prasarana yang mengacu kepada aktifitas pada proses bisnis *to-be* pelaporan keluhan sarana prasarana. Pada Tabel 7.17 berikut juga disertakan waktu terbaik, terburuk dan waktu rata-rata untuk menyelesaikan masing-masing aktifitasnya.

Tabel 7.17 Tabel Pengujian Waktu Aktifitas Pelaporan To-Be

No.	Aktor	Aktifitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pelapor	Melaporkan keluhan sarana prasarana	50 detik	100 detik	75 detik
2	Sistem	Mengirimkan notifikasi laporan	1 detik	3 detik	2 detik
3	Operator	Menerima laporan keluhan	10 menit	1 hari	30 menit
4	Operator	Mendatangi lokasi laporan keluhan	30 menit	2 jam	75 menit
5	Operator	Mencatat hasil pemeriksaan	20 menit	30 menit	25 menit
6	Operator	Menindaklanjuti keluhan	1 menit	3 menit	2 menit
7	Operator	Cek ketersediaan suku cadang/barang pengganti	10 detik	30 detik	20 detik
8	Operator	Cek ketersediaan teknisi	10 detik	30 detik	20 detik
9	Operator	Disposisi teknisi	10 detik	30 detik	20 detik
10	Teknisi	Melakukan perbaikan	1 jam	3 hari	2 jam
11	Teknisi	Melaporkan hasil perbaikan	30 menit	1 jam	45 menit
12	Operator	Mencatat hasil perbaikan	1 menit	10 menit	5 menit
Total Rata-Rata Waktu			2 jam 33 menit 21 detik	4 hari 3 jam 46 menit 13 detik	5 jam 4 menit 17 detik

Berdasarkan Tabel 7.17 tersebut diketahui bahwa untuk aktifitas pelaporan dari proses bisnis *to-be* didapatkan waktu penyelesaian terbaiknya memerlukan waktu 2 jam 33 menit 21 detik, waktu penyelesaian terburuknya memerlukan 4 hari 3 jam 46 menit 13 detik dan rata-rata dari waktu penyelesaiannya memerlukan 5 jam 4 menit 17 detik.

Terjadinya perbedaan waktu dari waktu penyelesaian terbaik dan terburuk dari proses bisnis *to-be* tersebut diakibatkan oleh faktor waktu dari aktifitas-aktifitas yang dilakukan oleh aktor Operator dan Teknisi. Waktu penyelesaian dari aktifitas Menerima Laporan Keluhan yang dilakukan oleh aktor Operator bergantung kepada kapan Operator memiliki waktu untuk mengerjakan aktifitas tersebut, apabila Operator tersebut sedang melakukan aktifitas Mendatangi lokasi laporan keluhan untuk melakukan survey atau Operator tersebut sedang melakukan

pekerjaan lain maka laporan keluhan yang diterima oleh sistem dapat mengalami keterlambatan penanganan. Sedangkan waktu penyelesaian dari aktifitas Melakukan perbaikan juga bergantung kepada ketersediaan suku cadang, barang pengganti atau ketersediaan dari teknisi yang diberikan disposisi. Selain itu waktu penyelesaian dari aktifitas ini juga bergantung kepada bagaimana pengerjaan dari perbaikan yang dilakukan.

Berikut pada Tabel 7.18 menunjukkan hasil pengujian waktu untuk aktifitas-aktifitas pelacakan status pelaporan keluhan sarana prasarana yang mengacu kepada aktifitas pada proses bisnis *to-be*. Berdasarkan Tabel 7.18 tersebut diketahui bahwa untuk aktifitas pelacakan dari proses bisnis *to-be* didapatkan waktu penyelesaian terbaiknya memerlukan waktu 2 detik, waktu penyelesaian terburuknya memerlukan 8 detik dan rata-rata dari waktu penyelesaiannya memerlukan 14 detik. Perbedaan waktu ini dapat terjadi ketika pelapor perlu memilih terlebih dahulu laporan keluhan mana yang ingin dilihat informasi lengkapnya.

Tabel 7.18 Tabel Pengujian Waktu Aktifitas Pelaporan To-Be

No.	Aktor	Aktifitas Yang Dikerjakan	Waktu		
			Terbaik	Terburuk	Rata-Rata
1	Pelapor	Melacak status laporan yang sedang dikerjakan	2 detik	8 detik	5 detik
2	Sistem	Menampilkan daftar laporan dan status pelaporan	1 detik	3 detik	2 detik
Total Rata-Rata Waktu			3 detik	11 detik	7 detik

7.3.2 Analisis Perbandingan Waktu

Pada Tabel 7.19 menunjukkan tabel hasil perbandingan waktu antara aktifitas-aktifitas proses bisnis *as-is* dengan aktifitas-aktifitas proses bisnis *to-be*. Aktifitas yang akan dibandingkan antara waktu *as-is* dengan waktu *to-be*nya adalah aktifitas untuk pelaporan dan pelacakan status. Waktu *as-is* dengan waktu *to-be* yang dijelaskan terdiri dari waktu terbaik, waktu terburuk dan waktu rata-ratanya untuk penyelesaian aktifitas.

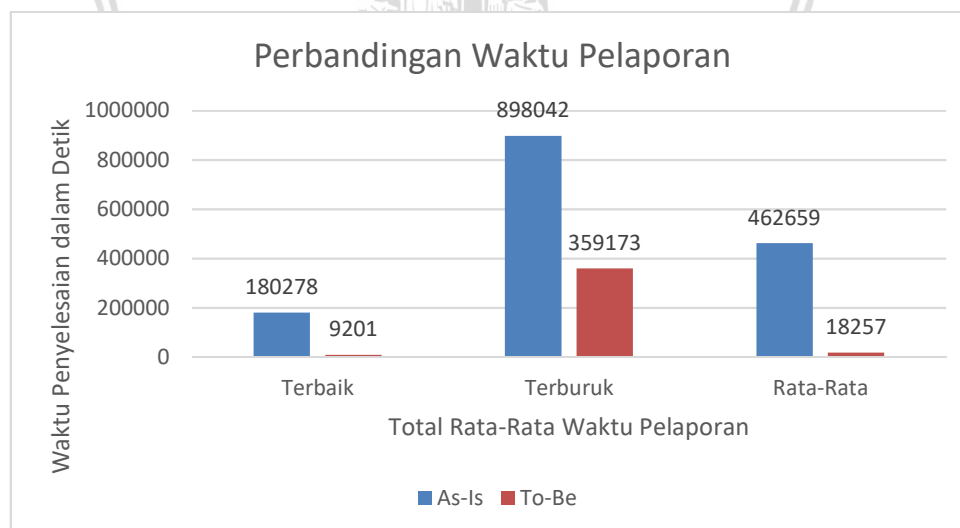
Tabel 7.19 Tabel Hasil Perbandingan Waktu

Proses Bisnis	Total Rata-Rata Waktu Pelaporan			Total Rata-Rata Waktu Pelacakan		
	Terbaik	Terburuk	Rata-Rata	Terbaik	Terburuk	Rata-Rata
As-Is	2 hari 2 jam 4 menit 38 detik	10 hari 9 jam 27 menit 22 detik	5 hari 8 jam 30 menit 59 detik	9 detik	21 detik	15 detik

To-Be	2 jam 33 menit 21 detik	4 hari 3 jam 46 menit 13 detik	5 jam 4 menit 17 detik	3 detik	11 detik	7 detik
Selisih	1 hari 23 jam 31 menit 17 detik	6 hari 5 jam 41 menit 9 detik	5 hari 3 menit 26 menit 42 detik	6 detik	10 detik	8 detik

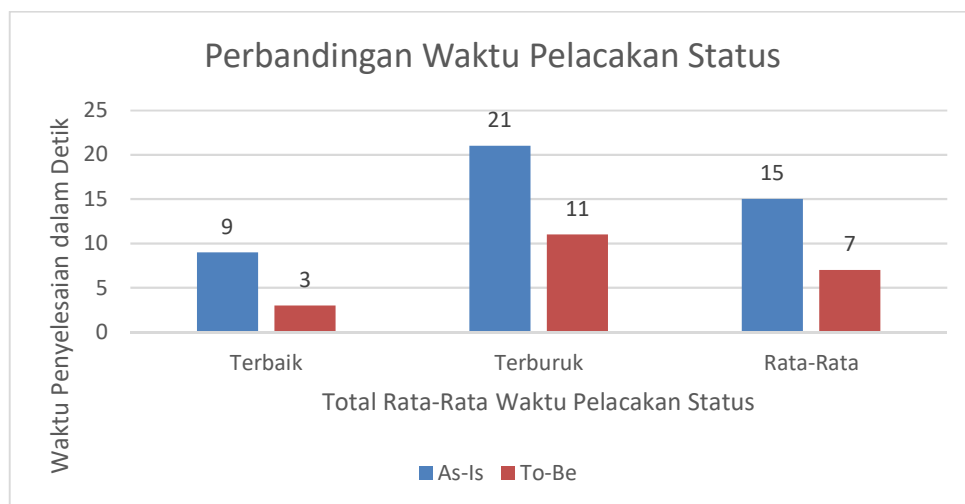
Berdasarkan Tabel 7.19 tersebut diketahui bahwa untuk waktu dari aktifitas pelaporan proses bisnis *as-is* dibandingkan dengan waktu dari aktifitas pelaporan proses bisnis *to-be* didapatkan selisih waktu sebanyak 1 hari 23 jam 31 menit 17 detik untuk waktu penyelesaian terbaiknya, selisih 6 hari 5 jam 41 menit 9 detik untuk waktu penyelesaian terburuknya dan selisih 5 hari 3 menit 26 menit 42 detik untuk waktu rata-ratanya. Sedangkan untuk waktu dari aktifitas pelacakan status pelaporan proses bisnis *as-is* dibandingkan dengan waktu dari aktifitas pelacakan status pelaporan proses bisnis *to-be* didapatkan selisih waktu sebanyak 6 detik untuk waktu penyelesaian terbaiknya, selisih 10 detik untuk waktu penyelesaian terburuknya dan selisih 8 detik untuk waktu rata-ratanya.

Berikut pada Gambar 7.5 disajikan grafik batang untuk mempermudah dalam pembacaan hasil perbandingan total rata-rata waktu pelaporan dari Tabel 7.19. Pada sumbu x menjelaskan waktu terbaik, waktu terburuk dan waktu rata-ratanya untuk penyelesaian aktifitas. Sedangkan pada sumbu y menjelaskan waktu yang dibutuhkan untuk menyelesaikan aktifitas pelaporan dan ditampilkan dalam satuan detik.



Gambar 7.5 Grafik Batang Perbandingan Waktu Pelaporan

Sedangkan untuk mempermudah dalam pembacaan hasil perbandingan total rata-rata waktu pelacakan status dari Tabel 7.19, berikut disajikan grafik batang pada Gambar 7.6. Pada sumbu x menjelaskan waktu terbaik, waktu terburuk dan waktu rata-ratanya untuk penyelesaian aktifitas pelacakan status. Sedangkan pada sumbu y menjelaskan waktu yang dibutuhkan untuk menyelesaikan aktifitas dan ditampilkan dalam satuan detik.



Gambar 7.6 Grafik Batang Perbandingan Waktu Pelacakan Status

Berdasarkan data waktu yang dipaparkan pada Tabel 7.19, Gambar 7.5 dan Gambar 7.6, dapat diketahui bahwa hasil pengujian waktu proses dari aktifitas proses bisnis *to-be* dapat diselesaikan lebih cepat dibandingkan dengan waktu proses dari aktifitas proses bisnis *as-is*.

BAB 8 KESIMPULAN DAN SARAN

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, kesimpulan yang dapat diambil dari pengembangan dan pengujian Sistem Informasi Manajemen Pelaporan Sarana dan Prasarana antara lain adalah sebagai berikut:

1. Berdasarkan hasil evaluasi proses bisnis yang dilakukan, ditemukan masalah yang terjadi pada pelaporan keluhan secara langsung saat ini adalah pelaporan secara lisan kepada Pegawai Perlengkapan membutuhkan waktu 4 sampai 10 menit. Pihak pelapor dan civitas akademik lain juga kesusahan untuk mendapatkan status pelaporan mengenai sarana prasarana yang telah dilaporkan. Kemudian masalah yang terjadi pada pelaporan keluhan melalui *E-Complaint* UB saat ini adalah pelaporan keluhan melalui *website E-Complaint* tidak tersedia layanan untuk mempermudah identifikasi barang sarana prasarana yang bermasalah. Proses pelacakan status pelaporan mengharuskan pelapor untuk memasukkan kode tiket keluhan sehingga berpotensi untuk terjadi pelaporan yang berulang oleh civitas lain. Solusi yang dapat dilakukan adalah membangun sistem yang dapat digunakan untuk melaporkan keluhan sarana prasarana, dapat mempermudah identifikasi barang sarana prasarana, dapat menampilkan status pelaporan dan laporan keluhan tersebut dapat langsung diterima oleh Pegawai Perlengkapan.
2. Berdasarkan hasil analisis kebutuhan sistem untuk Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana didapatkan kebutuhan fungsional sistem untuk aplikasi *android* pelapor antara lain adalah Kirim Laporan Keluhan, Lini Masa Laporan Keluhan, Pelacakan Laporan Keluhan dan Riwayat Laporan Keluhan. Sedangkan kebutuhan fungsional sistem untuk *website operator* antara lain adalah Terima Laporan Keluhan, Tindak Lanjut Keluhan, Kelola Barang, Kelola Teknisi dan Cek Daftar Pelapor. Kemudian kebutuhan tersebut dimodelkan ke dalam *use case diagram* dan untuk urutan scenario dari masing-masing *use case* dijelaskan dengan menggunakan *use case scenario*.
3. Hasil perancangan sistem untuk Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana diawali dengan perancangan arsitektur sistem, kemudian perancangan *sequence diagram*, perancangan *class diagram*, perancangan skema basis data yang mengacu kepada *class diagram*, perancangan antarmuka pengguna dan perancangan algoritme. Perancangan yang telah dilakukan dibagi menjadi aktor pelapor dan *operator* sesuai identifikasi aktor sebelumnya.

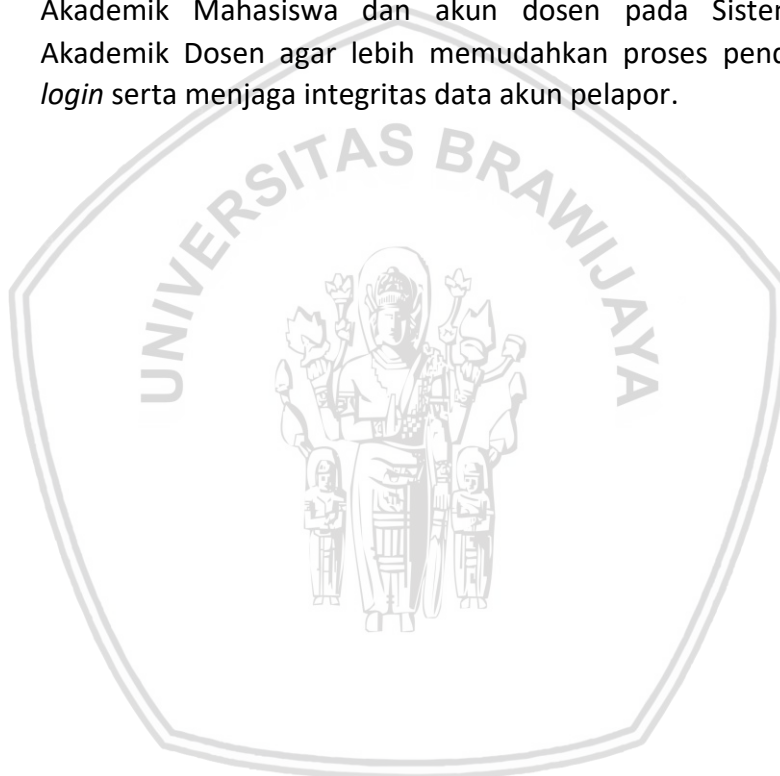
4. Hasil implementasi sistem untuk Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana adalah dengan mengembangkan aplikasi *mobile* pelapor yang dirancang dan dibangun menggunakan bahasa pemrograman Java dengan platform Android serta *website operator* yang dirancang dan dibangun menggunakan bahasa pemrograman PHP dengan menerapkan framework Codeigniter. Aplikasi *mobile* pelapor memiliki fitur utama untuk mengirim laporan yang disertai dengan fungsi kamera dan pindai kode QR, melihat status pelaporan keluhan beserta melihat rincian lengkap status pelaporan keluhan. Sedangkan *website operator* memiliki fitur utama untuk menerima dan melihat laporan, menindaklanjuti laporan keluhan, mengelola barang dan mengelola teknisi.
5. Berdasarkan pengujian untuk Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana yang dilakukan dengan menggunakan pengujian validasi didapatkan hasil 100% valid dan dapat disimpulkan bahwa perangkat lunak yang dikembangkan dapat berjalan sesuai dengan kebutuhan pengguna dan perancangan yang telah dibuat. Kemudian dilanjutkan dengan pengujian *usability* dengan metrik pengukuran *effectiveness* yang menghasilkan rata-rata 97,14% untuk pelapor dan 100% untuk *operator* yang berarti *task* yang diberikan dapat diselesaikan dengan baik oleh responden dan metrik *efficiency* yang menghasilkan angka TBE 0,198 *goals/sec*. Selanjutnya dari metrik *satisfaction* didapatkan hasil bahwa responden sangat setuju dengan sistem yang dikembangkan dapat memudahkan pelaporan dan pelacakan status bagi pelapor serta bagi *operator* untuk menerima dan menindaklanjuti laporan. Pengujian terakhir dilakukan pengujian perbandingan waktu yang mana setelah dilakukan perbandingan waktu proses bisnis *as-is* dengan proses bisnis *to-be* didapatkan hasil bahwa aktifitas pelaporan dan pelacakan status pelaporan pada proses bisnis *to-be* menggunakan sistem yang dikembangkan dapat diselesaikan dengan lebih cepat dibandingkan pada proses bisnis *as-is*.

8.2 Saran

Saran yang dapat diberikan kepada peneliti atau pengembang selanjutnya untuk melakukan pengembangan lanjut pada Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana di antaranya adalah:

1. Perlu dilakukan penelitian lebih lanjut mengenai kesiapan dan kelayakan dari Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana terhadap infrastruktur yang dimiliki oleh Fakultas Ilmu Komputer Universitas Brawijaya untuk mengetahui tingkat kesiapan dan kelayakan implementasi dari sistem informasi yang diajukan.

2. Perlu dilakukan pengembangan lanjut mengenai fungsi-fungsi lain yang perlu disediakan pada *Website Operator* Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana untuk lebih mendukung fungsi pengadaan dan pemeliharaan barang sarana prasarana.
3. Perlu dilakukan pengembangan pada antarmuka Sistem Informasi Manajemen Pelaporan Sarana Dan Prasarana baik pada aplikasi pelapor maupun pada *website operator* untuk memperbaiki antarmuka dan meningkatkan pengalaman pengguna agar sistem yang diajukan dapat lebih mudah untuk digunakan. Selain itu juga perlu dilakukan pengembangan lebih lanjut untuk aplikasi pelapor agar dapat terintegrasi dengan akun mahasiswa pada Sistem Informasi Akademik Mahasiswa dan akun dosen pada Sistem Informasi Akademik Dosen agar lebih memudahkan proses pendaftaran dan *login* serta menjaga integritas data akun pelapor.



DAFTAR PUSTAKA

- Abeyasinghe, S., 2008. *RESTful PHP Web Service*. Birmingham: Packt Publishing.
- Arroqui, M., Mateos, C., Machado, C. & Zunino, A., 2012. RESTful Web Services improve the efficiency of data transfer of a whole-farm simulator accessed by Android smartphones. *Computers and Electronics in Agriculture*, p. 5.
- Ashford, R., 2010. QR Codes and Academic Libraries Reaching Mobile Users. *College & Research Libraries News*, Volume 71, p. 5.
- Bertram, D., 2016. *Likert Scale*. s.l.:s.n.
- Bhardwaj, S., Sharma, R., Chouhan, P. & Sharma, P., 2013. Android Operating System. *International Journal of Engineering Technology & Management Research*, Volume 1, p. 4.
- Bisio, I., Sciarrone, I. & Zappatore, S., 2016. A new asset tracking architecture integrating RFID, Bluetooth Low Energy tags and ad hoc smartphone applications, Pervasive and Mobile Computing. *ScienceDirect*, p. 26.
- Booch, G., Rumbaugh, J. & Jacobson, I., 2005. *The Unified Modeling Language User Guide*. 2nd penyunt. Reading: Addison-Wesley Longman.
- Brandenburg, L., 2017. *Bridging The Gap*. [Online] Tersedia di: <http://www.bridging-the-gap.com/as-is-business-process/> [Diakses 30 January 2018].
- Cagle, R. F., 2003. Infrastructure Asset Management: An Emerging Direction. *AACE International Transaction*, p. 6.
- Choizes, 2017. *Diedit*. [Online] Tersedia di: <https://www.diedit.com/skala-likert/> [Diakses 7 May 2018].
- Deitel, P. & Deitel, H., 2012. *Java for Programmers*. 2nd penyunt. Boston: Pearson Education.
- Denso ADC, 2011. *QR Code Essentials*. s.l.:Denso ADC.
- Fetrina, E., Rustamaji, E., Nuraeni, T. & Durrachman, Y., 2017. Inventory Management Information System Development At BPRTIK Kemkominfo Jakarta. *IEEE*, p. 4.
- Given, L. M., 2008. *The SAGE Encyclopedia of Qualitative Research Methods*. 1 & 2 penyunt. London: SAGE Publications Inc..
- Hall, J. A., 2008. *Accounting Information System*. 6th penyunt. Mason: Cengage Learning.
- Hambling, B. & Goethem, P. v., 2013. *User Acceptance Testing: A Step-by-step Guide*. Swindon: BCS Learning & Development Limited.
- Hastings, N. A. J., 2010. *Physical Asset Management*. Brisbane: Springer.

- Hustinawati, Himawan, A. K. & Latifah, 2014. Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications*, Volume 94, p. 6.
- IBM, 2018. *Mapping Class Diagrams to Data Models*. [Online] Tersedia di: https://www.ibm.com/support/knowledgecenter/en/SS6RBX_11.4.3/com.ibm.sa.oomethod.doc/topics/c_create_LogicalPhysical_Diagr.html [Diakses 25 Februari 2018].
- IBM, 2018. *Physical Data Models*. [Online] Tersedia di: https://www.ibm.com/support/knowledgecenter/SS62YD_4.1.1/com.ibm.datatools.core.ui.doc/topics/cphysmod.html [Diakses 25 Februari 2018].
- Jogiyanto, H. M., 2008. *Metodologi Penelitian Sistem Informasi*. Yogyakarta: Andi Offset.
- Justin, M., 2012. Quantifying the Benefits of Asset Management. *Asset Management Consulting Limited*, p. 6.
- Khan, G. et al., 2016. Development of Transportation Asset Management and Data Collection System (TAMS) using Mobile Applications. *ScienceDirect*, p. 7.
- Kurniawan, T. A., 2018. Pemodelan Use Case (UML) Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi dan Ilmu Komputer*, p. 10.
- Lazar, J., Feng, J. H. & Hochheiser, H., 2017. *Research Methods in Human-Computer Interaction*. 2nd penyunt. Cambridge: Morgan Kaufmann.
- Lorenzi, D. et al., 2014. Enhancing The Government Service Experience Through QR Codes On Mobile Platforms. *Government Information Quarterly*, Volume 6, p. 11.
- Mehta, B. R. & Reddy, Y. J., 2014. *Industrial Process Automation Systems Design and Implementation*. s.l.:Butterworth-Heinemann.
- Mifsud, J., 2015. *Usabilitygeek*. [Online] Tersedia di: <https://usabilitygeek.com/usability-metrics-a-guide-to-quantify-system-usability/> [Diakses 9 Mei 2018].
- Mishra, A. & Dubey, D., 2013. A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios. *International Journal of Advance Research in Computer Science and Management Studies*, p. 6.
- Munassar, N. M. A. & Govardhan, A., 2010. A Comparison Between Five Models Of Software Engineering. *International Journal of Computer Science Issues*, p. 8.

- Object Management Group, 2011. *Business Process Model and Notation*. s.l.:Object Management Group.
- Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th penyunt. New York: McGraw-Hill.
- Ragunath, P. K. et al., 2010. Evolving A New Model (SDLC Model-2010) For Software Development Life Cycle (SDLC). *IJCSNS International Journal of Computer Science and Network Security*, p. 8.
- Santos, T. & Serrao, C., 2016. Secure Javascript Object Notation (SecJSON). *International Conference for Internet Technology and Secured Transaction*, p. 6.
- Saunders, M., Lewis, P. & Thornhill, A., 2009. *Research Methods for Business Students Ffth Edition*. 5th penyunt. Harlow: Pitman Publishing.
- Sauro, J., 2012. *Measuringu*. [Online] Tersedia di: <https://measuringu.com/seq10/> [Diakses 9 Mei 2018].
- Sommerville, I., 2007. *Software Engineering*. 8 penyunt. Harlow: Pearson.
- Sommerville, I., 2011. *Software Engineering*. 9th penyunt. New York: Addison-Wesley.
- Wazlawick, R. S., 2014. *Objeck-Oriented Analysis and Design for Information Systems*. Waltham: Morgan Kaufmann.

LAMPIRAN A HASIL WAWANCARA PEGAWAI PERLENGKAPAN

A.1 Wawancara Pegawai Perlengkapan

Wawancara ini bertujuan untuk mengetahui latar belakang masalah mengenai pelaporan sarana prasarana yang ada pada Fakultas Ilmu Komputer serta mengetahui proses dari pelaporan keluhan sarana prasarana. Wawancara dilakukan dengan pegawai perlengkapan Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya pada :

Waktu : 1 November 2017
 Tempat : Lantai 6 Gedung F FILKOM UB
 Narasumber : Felix Panji Andrianto
 Jabatan : Pegawai Bagian Umum dan Perlengkapan

Hasil Wawancara

Keterangan: P = Pewawancara N = Narasumber

P : Bagaimana proses pelaporan keluhan mengenai sarana prasarana di FILKOM yang berjalan saat ini?

N : Pelaporan keluhan sarana prasarana yang ada di wilayah FILKOM saat ini dapat dilakukan oleh civitas akademik FILKOM dengan beberapa cara. Cara yang pertama adalah dengan melaporkan keluhan tersebut kepada pihak pegawai perlengkapan secara langsung. Selain menyampaikan secara langsung, beberapa civitas akademik FILKOM terutama bagi dosen dan pegawai lain yang sudah memiliki nomor telepon dari pegawai perlengkapan juga dapat menyampaikan keluhannya melalui telepon, SMS atau *Whatsapp*. Beberapa mahasiswa juga terkadang ada yang menyampaikan melalui cara tersebut, tetapi kebanyakan masih segan untuk menyampaikannya secara langsung atau pun menghubungi melalui media komunikasi. Selain kedua cara tersebut, civitas akademik juga dapat melaporkan keluhan sarana prasarananya melalui *Website E-Complaint* Universitas Brawijaya, laporan dari *website* tersebut juga kami terima dan tindak-lanjuti.

P : Bagaimana proses penindaklanjutan keluhan mengenai sarana prasarana tersebut oleh pegawai perlengkapan FILKOM?

N : Setelah keluhan sarana prasarana telah kami terima, maka dari pegawai perlengkapan akan melakukan survey ke lokasi untuk memeriksa dan mencatat keadaan dari sarana prasarana tersebut. Kemudian apabila sarana prasarana tersebut perlu dilakukan perbaikan atau penggantian maka akan kami periksa terlebih dahulu mengenai ketersediaan suku cadangnya atau barang penggantinya. Jika suku cadang atau barang pengganti sudah tersedia dan teknisi juga tersedia akan segera diperbaiki oleh teknisi, sedangkan jika suku cadangnya atau barang penggantinya tidak tersedia terpaksa harus menunggu pengadaan terlebih dahulu.

P : Apakah sekiranya ada kendala dari proses penindaklanjutan keluhan mengenai sarana prasarana tersebut?

N : Terkadang yang menjadi kendala untuk penindaklanjutan keluhan kenapa menjadi lama salah satunya adalah menunggu ketersediaan dari barang yang sedang dalam proses pengadaan dan ketersediaan teknisi yang mungkin masih mengerjakan pekerjaan lain

atau contohnya seperti kerusakan *Air Conditioner* maka menunggu teknisi dari vendor datang. Sedangkan civitas akademik terkadang tidak mengetahui dan tidak menyadari mengenai faktor-faktor tersebut, sehingga terkadang mengganggu kinerja dari pegawai perlengkapan kurang tanggap.

P : Sebelumnya disampaikan mengenai civitas akademik juga dapat melaporkan keluhan melalui *website E-Complaint UB*, bagaimanakah proses penindaklanjutan keluhannya dari pegawai perlengkapan?

N : Untuk proses penindaklanjutan keluhan dari *website E-Complaint UB* kurang lebih sama, yang membedakan adalah pegawai perlengkapan mendapatkan laporan keluhan berupa surat disposisi dari pegawai sekretaris dekan karena *website E-Complaint UB* untuk FILKOM dikelola oleh sekretaris dekan tersebut. Setelah keluhan-keluhan diterima oleh sekretaris selanjutnya keluhan tersebut dipilah berdasarkan jenis keluhannya dan dibuatkan surat disposisi, kemudian untuk keluhan mengenai sarana prasarana akan diberikan kepada pegawai perlengkapan. Pada surat disposisi tersebut terdapat informasi mengenai masalah keluhannya beserta saran tindakannya dari pelapor. Setelah keluhan tersebut selesai diperbaiki pegawai perlengkapan memberikan surat balasan kepada sekretaris beserta tanggapan mengenai keluhan tersebut, yang oleh sekretaris akan dimasukkan kedalam *website E-Complaint UB*.

P : Apakah terdapat kendala dalam proses pelaporan keluhan sarana prasarana melalui *website E-Complaint UB* tersebut?

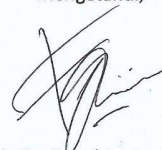
N : Kendala yang sering terjadi adalah waktu yang dibutuhkan untuk keluhan tersebut sampai ke pegawai perlengkapan biasanya cukup lama, rata-rata sampai 5 hari kerja atau bahkan hampir 2 minggu. Sedangkan di waktu tersebut pegawai perlengkapan sudah atau sedang menangani masalah yang terjadi. Tetapi sering kalinya sekitar 5-7 hari kerja saja.

P : Selain informasi mengenai masalah keluhan dan saran tindakan, apakah dalam pelaporan melalui *website E-Complaint UB* terdapat informasi tambahan lain seperti foto dari masalah keluhan yang dilaporkan?

N : Selama ini belum ada informasi tambahan seperti foto kejadian atau foto barang.

Malang, 1 November 2017

Mengetahui,



Ferix Panji Andrianto, S.ST

LAMPIRAN B HASIL WAWANCARA MAHASISWA CIVITAS FILKOM UB

B.1 Wawancara Mahasiswa 1

Wawancara ini bertujuan untuk mengetahui latar belakang masalah mengenai pelaporan sarana prasarana yang ada pada Fakultas Ilmu Komputer serta mengetahui proses dari pelaporan keluhan sarana prasarana. Wawancara dilakukan dengan mahasiswa civitas akademik Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya pada :

Waktu : 9 November 2017
Tempat : FILKOM UB
Narasumber : Tuti Wardani Hamid
Jabatan : Mahasiswi

Hasil Wawancara

Keterangan: P = Pewawancara N = Narasumber

P : Bagaimana proses pelaporan keluhan mengenai sarana prasarana di FILKOM yang pernah dilakukan selama ini?

N : Selama ini pelaporan keluhan mengenai sarana prasarana di FILKOM disampaikan secara lisan langsung kepada pegawai perlengkapan di Gedung F Lantai 6.

P : Apakah Anda sebagai pelapor setelah melakukan pelaporan keluhan pernah mendapatkan pemberitahuan/informasi mengenai tanggapan untuk laporan tersebut dari pegawai perlengkapan?

N : Selama ini jarang mendapatkan tanggapan apabila tidak di *follow-up* lagi secara langsung, kecuali apabila menggunakan sistem *E-Complaint* UB.

P : Keluhan seperti apa yang pernah Anda laporkan melalui sistem *E-Complaint* UB?

N : Keluhan mengenai banjir yang terjadi di Gedung D, kebocoran pada gedung F, stopkontak dan lampu yang mati di Gazebo Filkom.

P : Bagaimana cara Anda dalam melaporkan keluhan mengenai sarana prasarana tersebut melalui *E-Complaint* UB?

N : Pertama saya memasukkan penjelasan keluhan yang saya alami ke dalam *form* yang telah disediakan pada website tersebut, kemudian setelah *submit* akan mendapatkan nomor

kode keluhan. Selanjutnya apabila ingin memeriksa *progress complaint* kita maka bisa memasukkan kode keluhan tersebut kedalam website *E-Complaint*.

P : Selama menjelaskan dan memasukkan keluhan yang dialami, bagaimana cara Anda menunjukkan ke sistem tersebut barang mana yang sedang bermasalah?

N : Selama ini menggunakan penjelasan saja, belum disediakan fitur tautan gambar.

P : Berdasarkan pengalaman Anda, berapa waktu yang dibutuhkan untuk laporan keluhan Anda mendapatkan tanggapan?

N : Selama ini paling cepat sekitar 4-5 hari kerja dan paling lama sekitar 7 hari kerja, biasanya antara langsung diperbaiki atau di survey terlebih dahulu ke tempat kejadian.

Malang, 9 November 2017

Mengetahui,



Tuti Wardani Hamid

B.2 Wawancara Mahasiswa 2

Wawancara ini bertujuan untuk mengetahui latar belakang masalah mengenai pelaporan sarana prasarana yang ada pada Fakultas Ilmu Komputer serta mengetahui proses dari pelaporan keluhan sarana prasarana. Wawancara dilakukan dengan mahasiswa civitas akademik Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya pada :

Waktu : 10 November 2017
Tempat : FILKOM UB
Narasumber : Afifahra Aliif Rizromi
Jabatan : Mahasiswa

Hasil Wawancara

Keterangan: P = Pewawancara N = Narasumber

P : Bagaimana proses pelaporan keluhan mengenai sarana prasarana di FILKOM yang pernah dilakukan selama ini?

N : Selama ini melaporkan keluhan mengenai sarana prasarana di FILKOM melalui kuesioner yang disebar oleh perwakilan lembaga mahasiswa FILKOM, baik dari tingkat fakultas maupun jurusan yang nantinya kumpulan kuesioner tersebut disampaikan kepada pegawai perlengkapan di lantai 6. Selain itu juga terkadang melaporkan melalui sistem *E-Complaint UB*.

P : Apakah ada alasan mengapa Anda memilih melaporkan melalui kuesioner secara kolektif tersebut dibandingkan melaporkan secara personal?

N : Karena menurut saya cukup berat apabila setiap ingin melaporkan keluhan harus berjalan menuju Gedung F Lantai 6 terlebih dahulu. Selain itu apabila melaporkan dengan cara bertemu secara *personal* terkadang masih merasa segan dengan pegawai yang bertugas, sehingga apabila sedang tidak disebar kuesioner kolektif dari lembaga mahasiswa maka saya melaporkan melalui sistem *E-Complaint UB*.

P : Dalam melaporkan keluhan sarana prasarana melalui kuesioner secara kolektif, apakah pernah mendapatkan tanggapan mengenai tindak lanjut dari laporan keluhan tersebut?

N : Selama ini tidak mendapat informasi tanggapan mengenai keluhan tersebut, kecuali dari perwakilan pihak lembaga rutin melakukan *follow up* kepada pegawai perlengkapan.

P : Apakah menurut Anda informasi tanggapan mengenai keluhan yang telah dilaporkan oleh pelapor perlu diketahui oleh si pelapor maupun civitas akademik FILKOM lainnya?

N : Informasi tanggapan dari laporan keluhan tersebut penting dan pelapor maupun civitas akademik FILKOM lainnya perlu mendapatkan kepastian mengenai laporan tersebut sifatnya masih di proses, dikerjakan, ditunda atau ditolak. Apabila laporan tersebut ditunda atau ditolak juga perlu diketahui keterangan alasan dari penundaan atau penolakannya, dengan demikian maka aliran informasi mengenai pelaporan sarana prasarana dan tindak lanjutnya bisa lebih jelas dan transparan.

Malang, 10 November 2017

Mengetahui,



Afifahlya Alif Rizyomi

LAMPIRAN C HASIL KUESIONER TASK LEVEL SATISFACTION PELAPOR

C.1 Responden 1

Kuesioner Task Level Satisfaction
Pengguna Aplikasi *Mobile* Pelapor
Sistem Informasi Manajemen Aset Pelaporan Sarana Prasarana

Nama Responden	Yustinus Radityo Pradana
Jabatan Responden	Mahasiswa
Tanggal Pengujian	14 Mei 2018
Spesifikasi Perangkat	HTC One M8 Marshmallow

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai pelapor mudah untuk dilakukan?	TP-01				✓	
2	Apakah proses untuk melakukan <i>login</i> sebagai pelapor mudah untuk dilakukan?	TP-02					✓
3	Apakah proses untuk membuat dan mengirimkan laporan keluhan sarana prasarana mudah untuk dilakukan?	TP-03				✓	
4	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan yang sedang ditangani mudah untuk dilakukan?	TP-04				✓	

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan riwayat mudah untuk dilakukan?	TP-05			✓		
6	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan lini masa mudah untuk dilakukan?	TP-06					✓
7	Apakah proses untuk melakukan <i>logout</i> mudah untuk dilakukan?	TP-07				✓	

Responden

Yustinus Radityo P.

C.2 Responden 2

Kuesioner Task Level Satisfaction

Pengguna Aplikasi Mobile Pelapor

Sistem Informasi Manajemen Aset Pelaporan Sarana Prasarana

Nama Responden	Iqbal Taufiq A.N.
Jabatan Responden	Mahasiswa
Tanggal Pengujian	15 Mei 2018
Spesifikasi Perangkat	Asus Zenfone GO Lollipop

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai pelapor mudah untuk dilakukan?	TP-01				✓	
2	Apakah proses untuk melakukan login sebagai pelapor mudah untuk dilakukan?	TP-02					✓
3	Apakah proses untuk membuat dan mengirimkan laporan keluhan sarana prasarana mudah untuk dilakukan?	TP-03				✓	
4	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan yang sedang ditangani mudah untuk dilakukan?	TP-04					✓

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan riwayat mudah untuk dilakukan?	TP-05					✓
6	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan lini masa mudah untuk dilakukan?	TP-06					✓
7	Apakah proses untuk melakukan <i>logout</i> mudah untuk dilakukan?	TP-07					✓

Responden


(IQBAL)

C.3 Responden 3

Kuesioner Task Level Satisfaction

Pengguna Aplikasi Mobile Pelapor

Sistem Informasi Manajemen Aset Pelaporan Sarana Prasarana

Nama Responden	Ryan Dwi Pambudi
Jabatan Responden	Mahasiswa
Tanggal Pengujian	14 Mei 2018
Spesifikasi Perangkat	Alcatel Flash Plus Marshmallow

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai pelapor mudah untuk dilakukan?	TP-01				✓	
2	Apakah proses untuk melakukan login sebagai pelapor mudah untuk dilakukan?	TP-02					✓
3	Apakah proses untuk membuat dan mengirimkan laporan keluhan sarana prasarana mudah untuk dilakukan?	TP-03				✓	
4	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan yang sedang ditangani mudah untuk dilakukan?	TP-04				✓	

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan riwayat mudah untuk dilakukan?	TP-05					✓
6	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan lini masa mudah untuk dilakukan?	TP-06					✓
7	Apakah proses untuk melakukan <i>logout</i> mudah untuk dilakukan?	TP-07				✓	

Responden

(Ryan Pwi P.)

C.4 Responden 4

Kuesioner Task Level Satisfaction

Pengguna Aplikasi Mobile Pelapor

Sistem Informasi Manajemen Aset Pelaporan Sarana Prasarana

Nama Responden	Fajar Krisna Wicaksono
Jabatan Responden	Mahasiswa
Tanggal Pengujian	14 Mei 2018
Spesifikasi Perangkat	Samsung Galaxy S6 Nougat

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai pelapor mudah untuk dilakukan?	TP-01					✓
2	Apakah proses untuk melakukan login sebagai pelapor mudah untuk dilakukan?	TP-02					✓
3	Apakah proses untuk membuat dan mengirimkan laporan keluhan sarana prasarana mudah untuk dilakukan?	TP-03			✓		
4	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan yang sedang ditangani mudah untuk dilakukan?	TP-04					✓

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan riwayat mudah untuk dilakukan?	TP-05					✓
6	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan lini masa mudah untuk dilakukan?	TP-06					✓
7	Apakah proses untuk melakukan <i>logout</i> mudah untuk dilakukan?	TP-07					✓

Responden

Fajar
(Fajar Krisna W.)

C.5 Responden 5

Kuesioner Task Level Satisfaction

Pengguna Aplikasi Mobile Pelapor


Sistem Informasi Manajemen Aset Pelaporan Sarana Prasarana

Nama Responden	Tahmidya Kamila P. N.
Jabatan Responden	Mahasiswa
Tanggal Pengujian	19 Mei 2018
Spesifikasi Perangkat	HP HTC One M8

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai pelapor mudah untuk dilakukan?	TP-01			✓		
2	Apakah proses untuk melakukan login sebagai pelapor mudah untuk dilakukan?	TP-02					✓
3	Apakah proses untuk membuat dan mengirimkan laporan keluhan sarana prasarana mudah untuk dilakukan?	TP-03			✓		
4	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan yang sedang ditangani mudah untuk dilakukan?	TP-04					✓

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan riwayat mudah untuk dilakukan?	TP-05					✓
6	Apakah proses untuk mendapatkan dan melihat informasi mengenai laporan lini masa mudah untuk dilakukan?	TP-06					✓
7	Apakah proses untuk melakukan <i>logout</i> mudah untuk dilakukan?	TP-07					✓

Responden


(Tahmidya)

LAMPIRAN D HASIL KUESIONER TASK LEVEL SATISFACTION OPERATOR

D.1 Responden 1

Kuesioner Task Level Satisfaction

Pengguna Website Operator

Sistem Informasi Manajemen Aset Pelaporan Sarana Prasarana

Nama Responden	Ferix Panji Andrianto, S.ST
Jabatan Responden	Pegawai Rencanakuhan Bagian Pemeliharaan
Tanggal Pengujian	14 Mei 2018
Spesifikasi Perangkat	Google Chrome, Desktop View Website Operator

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1	Apakah proses untuk mendaftarkan akun baru sebagai operator mudah untuk dilakukan?	TO-01				✓	
2	Apakah proses untuk melakukan login sebagai operator mudah untuk dilakukan?	TO-02				✓	
3	Apakah proses untuk mendapatkan dan melihat notifikasi laporan keluhan sarana prasarana mudah untuk dilakukan?	TO-03			✓		
4	Apakah proses untuk mendapatkan dan melihat seluruh laporan keluhan sarana prasarana mudah untuk dilakukan?	TO-04				✓	

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
5	Apakah proses untuk merubah status laporan keluhan sarana prasarana dan melakukan disposisi teknisi mudah untuk dilakukan?	TO-05				✓	
6	Apakah proses untuk melakukan <i>logout</i> dari sistem mudah untuk dilakukan?	TO-06				✓	

Responden


(F691x)